Lattice Boltzmann Method and Immersed Boundary Method for the simulation of viscous fluid flows

Emmanouil Falagkaris

THE UNIVERSITY
of EDINBURGH

Thesis submitted in fulfilment of the requirements for the degree of
Doctor of Philosophy
to the University of Edinburgh
October 2018

Declaration

I declare that this thesis has been composed solely by myself and that it has not been submitted, either in whole or in part, in any previous application for a degree. Except where otherwise acknowledged, the work presented is entirely my own.

Emmanouil Falagkaris
October 2018

Abstract

Most realistic fluid flow problems are characterised by high Reynolds numbers and complex boundaries. Over the last ten years, immersed boundary methods (IBM) that are able to cope with realistic geometries have been applied to Lattice-Boltzmann methods (LBM). These methods, however, have normally been applied to low Reynolds number problems. In the present work, an iterative direct forcing IBM has been successfully coupled with a multi-domain cascaded LBM in order to investigate viscous flows around rigid, moving and wilfully deformed boundaries at a wide range of Reynolds numbers. The iterative force-correction immersed boundary method of (Zhang *et al.*, 2016) has been selected due to the improved accuracy of the computation, while the cascaded LB formulation is used due to its superior stability at high Reynolds numbers. The coupling is shown to improve both the stability and numerical accuracy of the solution. The resulting solver has been applied to viscous flow (up to a Reynolds number of 100000) passed a NACA-0012 airfoil at a 10 degree angle of attack. Good agreement with results obtained using a body-fitted Navier-Stokes solver has been obtained.

At moving or deformable boundary applications, emphasis should be given on the influence of the internal mass on the computation of the aerodynamic forces, focusing on deforming boundary motions where the rigid body approximation is no longer valid. Both the rigid body and the internal Lagrangian points approximations are examined. The resulting solver has been applied to viscous flows around an in-line oscillating cylinder, a pitching foil, a plunging SD7003 airfoil and a plunging and flapping NACA-0014 airfoil. Good agreement with experimental results and other numerical schemes has been obtained. It is shown that the internal Lagrangian points approximation accurately captures the internal mass effects in linear and angular motions, as well as in deforming motions, at Reynolds numbers up to $4 \cdot 10^4$.

Finally, an expanded higher-order immersed boundary method which addresses two major drawbacks of the conventional IBM will be presented. First, an expanded velocity profile scheme has been developed, in order to compensate for the discontinuities caused by the gradient of the velocity across the boundary. Second, a numerical method derived from the Navier-Stokes equations in order to correct the pressure distribution across the boundary has been examined. The resulting hybrid solver has been applied to viscous flows around stationary and oscillating cylinders and examined the hovering flight of elliptical wings at low Reynolds numbers. It is shown that the proposed scheme smoothly expands the velocity profile across the boundary and increases the accuracy of the immersed boundary method. In addition, the pressure correction algorithm correctly expands the pressure profile across the boundary leading to very accurate pressure coefficient values along the boundary surface.

The proposed numerical schemes are shown to be very efficient in terms of computational cost. The majority of the presented results are obtained within a few hours of CPU time on a 2.8 GHz Intel Core i7 MacBook Pro computer with a 16GB memory.

Lay Summary

A novel 2-dimensional numerical scheme for the solution of the flow around stationary and moving bodies is developed in the present thesis. The ability of the proposed algorithm to accurately capture the underlying physics is demonstrated by investigations of various industrial application flows, including highly unsteady flows around several airfoil/flow configurations. Emphasis is given on the aerodynamic behaviour of the immersed bodies, as well as on the efficient and accurate resolution of the vortex structures in the near wake. The examined cases produced valuable and insightful understanding of the complex flow characteristics and aerodynamic loads acting on 2-dimensional bodies. The presented numerical results have been extensively validated, showing very good agreement with other numerical and experimental results.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor Prof. David Ingram for the excellent collaboration during my Ph.D studies, his continuous support, motivation and knowledge.

I would also like to thank the rest of my supervisory team: Dr. Ignazio Maria Viola and Dr. Daniel Friedrich for their encouragement and insightful advice throughout the last years.

My sincere thanks also goes to Dr. Timm Krügger for his useful advice and guidance at the early stages of my research.

I would also like to thank my dear friend and colleague, Dr. Ioannis Bitharas for the continuous support, encouragement, and useful discussions.

Special thanks should go to my friends and colleagues, Alexandros Kleidaras and Konstantinos Markakis for the continuous support, useful discussions and excellent collaboration.

I would also like to thank my partner and best friend Maria Melissourgou, as well as her parents, Iakovos and Vaso for their moral support.

I would also like to express my sincere gratitude to my uncle Prof. George Roussas for his continuous support, encouragement and motivation over my academic studies.

Last but not the least, I would like to thank my family: my parents John Falagkaris and Anna Georgiadou and my beloved brother Dimitris for supporting me spiritually throughout my academic studies and my life in general.

Finally, i wish to acknowledge funding from the Energy Technology Partnership (ETP) in support of my PhD studies [ETP125]. The ETP is a pooling exercise funded by the Scottish Funding Council and Scottish Enterprise.

Contents

List of Tables

List of Figures

List of Algorithms

Nomenclature

$\boldsymbol{\Omega}_a$	Collision operator
δ	Boundary layer thickness
ϵ	Internal energy
ν	Kinematic viscosity
ω	vorticity
ω_α	Relaxation parameter
ρ	Density
τ	Relaxation time
$\mathbf{c}$	Microscopic velocity
$\mathbf{F}$	Eulerian body force
$\mathbf{G}_{\mathscr{L}}$	Lagrangian body force
$\mathbf{T}_{\mathscr{L}}$	Lagrangian torque
$\mathbf{u}$	Velocity
$\mathbf{U}_{\mathscr{L}}$	Lagrangian velocity
$\mathbf{x}$	Eulerian grid coordinates
$\mathbf{X}_{\mathscr{L}}$	Lagrangian coordinates
$\widehat{\mu}_{x^m y^n}$	Continuous raw moments of f
$\widehat{k}_{x^m y^n}$	Discrete central moments of f
$\widehat{M}_{x^m y^n}$	Continuous central moments of f
$\widehat{m}_{x^m y^n}$	Discrete raw moments of f
C_D	Drag coefficient

C_f	Skin friction coefficient
C_L	Lift coefficient
C_P	Pressure coefficient
c_s	Lattice speed of sound
D	Physical dimension
e_α	Discrete velocity set
f	Particle mass distribution function
f_d	Vortex shedding frequency
f_α	Discrete distribution function
$f_\alpha^{(eq)}$	Equilibrium distribution function
$f_\alpha^{(neq)}$	Non-equilibrium distribution function
$f^{\mathscr{M}}$	Continuous Maxwell Boltzmann distribution function
h_i	Grid spacing
k_B	Boltzmann constant
m	Particle mass
Ma	Mach number
p_∞	Freestream pressure
Re	Reynolds number
S_α	Discrete source term
St	Strouhal number
T	Temperature
u_∞	Freestream velocity

Abbreviations

BC Boundary Condition

BGK Bhatnagar-Gross-Krook

CFD Computational Fluid Dynamics

CLBM Cascaded Lattice Boltzmann Method

DoF Degrees of Freedom

EDM Exact Difference Method

IBM Immersed Boundary Method

IDF Iterative Direct Forcing

LBM Lattice Boltzmann Method

LGCA Lattice Gas Cellular Automata

LPA Lagrangian Point Approximation

MDF Multi Direct Forcing

MRT Multiple Relaxation Time

NSE Navier Stokes Equations

PCA Pressure Correction Algorithm

PDF Particle Distribution Function

RBA Rigid Body Approximation

SRT Single Relaxation Time

TRT Two Relaxation Time

Chapter 1

Introduction

1.1 Aims and Objectives

The original focus of the present PhD project was the development of a numerical scheme in order to simulate complex, unsteady flows around vertical axis wind turbines. In vertical axis wind turbine simulations the flow is characterised by high Reynolds numbers, vortex shedding and complex rotational motions of the blades. This is a very challenging issue in terms of numerical modelling.

The biggest challenge and aim of the project was the development of a stable, highly accurate and efficient 2-Dimensional numerical scheme able to address the aforementioned flow characteristics and produce valuable and insightful results.

The objectives of the projects were the following.

1. Development of a stable and accurate fluid flow solver for high Reynolds numbers.
2. Development of a flexible, accurate and efficient solid solver.
3. Efficient coupling of the fluid and solid solvers.

4. Avoid complex meshing techniques.
5. Achieve high resolution accuracy around the investigated flow regions.
6. Understand the complex physics of fluid flows around moving boundaries at high Reynolds numbers.
7. Predict the aerodynamic behaviour of different airfoil designs. That includes the aerodynamic forces and the pressure distribution around the foil, as well as the boundary layer characteristics in unsteady vortex shedding configurations.
8. Estimate changes in the aerodynamic behaviour of the foil by imposing prescribed deformations.
9. Efficient resolution of vortex structures in the wake.

It will be demonstrated in Chapters 6, 7 and 8 that the aforementioned objectives have been extensively and successfully addressed. It should be noted that all numerical simulations presented on this thesis have been performed on a 2.8 GHz Intel Core i7 MacBook Pro computer with a 16GB memory. With a proper parallel implementation and some modifications and additions to the implemented physics the algorithm can be extended to 3 dimensions.

1.2 State of the Art

1.2.1 The Lattice-Boltzmann method

The lattice Boltzmann method (LBM), based on minimal discrete kinetic models, has been developed into a promising numerical scheme for simulating various fluid mechanics problems [8, 9, 10, 11] and other physical phenomena. In the field of computational fluid dynamics (CFD) the LBM has been well established as a successful solver for the weakly compressible Navier-Stokes equations [12, 13], as well as other, more complex physical problems including multiphase and multicomponent flows [14, 15]. Originating from the lattice gas cellular automata (LGCA) [16] (and more specifically the FHP model named after Frisch, Hasslacher and Pomeau), the LBM has undergone several advancements in order to remove the statistical noise of the LGCA and improve the numerical accuracy. In the LGCA, the flow is modelled using fictitious particles moving along discrete lattice directions and colliding under some simple rules. In order to remove the statistical noise of the LGCA caused by the discreteness of the particles, McNamara and Zanetti [17] replaced the particles by their velocity distribution functions. Following the work of McNamara and Zanetti [17], Higuera and Jimenez [18] presented the first lattice Boltzmann method.

The LBM may be realised as a simplified approximation of the continuous Boltzmann equation that models the statistical properties of particle collisions and propagation in a two-step stream and collide process. The streaming may be realised as a Lagrangian free process where particle populations move freely along discrete lattice directions, whereas collision may be described as a relaxation process towards a local equilibrium. Both steps are constructed in such a way that certain symmetry and conservation rules are satisfied. The macroscopic dynamics are then recovered using the averaged behaviour of the evolution of the particle populations. The choice of a suitable collision model can significantly affect the fidelity and stability of a LBM scheme. Therefore, the majority of the LBM advancements focus mainly on the construction of admissible collision operators. It has been found that due to the exact conservation in the streaming process and a round-off conservation in the collision step, the LBM, as a second-order accurate scheme, has significantly lower numerical dissipation compared to other second-order numerical methods [19]. The accuracy and simplicity of the LBM have established the method as a very good alternative to classical finite difference, finite volume, finite element and spectral element solvers [20, 21, 22, 23].

There exist a few methods that establish the consistency of the lattice-Boltzmann method (LBM) with regard to the Navier-Stokes equations (NSE). The multiscale expansion of Chapman and Cowling [24] is widely used to validate the consistency of a LBM numerical scheme [12, 13]. More recently, Sone [25] and Junk *et al.* [26] presented an asymptotic analysis based on the Hilbert expansion, whereas, Asinari [27] used a procedure based on the Grad moment expansion [28]. The asymptotic analysis [26] shows that the evolution of some observable quantities of the LBM does not affect the asymptotic behavior of the method [29]. Therefore, the scattering operator can be chosen somehow arbitrarily as long as the required constraints are obeyed. However, the number of constraints, including invariance under collision, translation and rotation, is not enough to fix all degrees of freedom resulting in a wide range of admissible operators.

Chen *et al.* [30] and Qian *et al.* [31] proposed the most common, single-relaxation-time (SRT) collision operator, which is based on the Bhatnagar-Gross-Krook (BGK) approximation [32]. The main idea of the BGK models is that in the collision process the velocity distributions relax towards a local equilibrium (Maxwellian) at a prescribed characteristic constant relaxation time. He and Luo [33] derived a numerical scheme from the continuous BGK modelusing a proper discretisation of the velocity space. The slight error in the derivation was later fixed by He *et al.* [34]. In the work of Shan *et al.* [35], the discretisation of the velocity space was done using Hermite polynomials and Gauss-Hermite quadrature rules.

The BGK collision model, though very popular due to its simplicity, has certain limitations but most importantly is prone to numerical instabilities at higher Reynolds numbers. In order to address the aforementioned issues and to enhance the stability of the collision step, d'Humières [36] heuristically presented a multiple-relaxation-time (MRT) operator based on the raw moment formulation of the distribution functions. The collisions are performed in moment space, where each moment is allowed to relax towards it's equilibrium state at a different rate. Thus, by carefully treating each relaxation time separately, the numerical stability is significantly improved [37]. In addition, Ginzburgh and d'Humières [38] showed that the physical representation of certain problems can be further improved by imposing specific relaxation rates at certain non-hydrodynamics moments. However, it should be noted that this MRT formulation [36] is not unique [39]. Luo *et al.* [40] presented a detailed comparison between the MRT scheme and other collision models. An intermediate version of the MRT and the SRT chemes is the two-relaxation-time (TRT) model, in which even and odd order moments relax towards their equilibrium states at different rates [41]. In both TRT and MRT models the relaxation rates can be adjusted in such a way that any undesirable discrete kinetic effects at certain flow conditions near the walls can be suppressed.

Another class of collision operators was developed [42, 43] based on a minimization solution of Boltzmann's H-function. In the so-called entropic LBM, collision is modulated with a single relaxation time by enforcing local involution of entropy (H-theorem), aiming at maintaining non linear stability. The collision operator in the entropic LBM differs from the ones found in the SRT or MRT models as it requires non-polynomial functions of the hydrodynamic variables. Asinari and Karlin [44] presented a novel entropy based MRT model. Prasianakis *et al.* [45] developed an entropic LBM where the Galilean invariance of the scheme is restored. Karlin *et al.* [177] presented a series of entropy functions whose local equilibria are suitable to recover the NSE in the framework of the LBM. Frapolli *et al.* [178] presented an entropic lattice Boltzmann model for compressible flows. In the work of Mazloomi *et al.* [179] thermodynamically consistent entropic lattice Boltzmann model for multi-phase flows has been derived. Extended entropic LBM including properly implemented boundary conditions may be found in [180, 181, 182].

Most recently, Geier *et al.*, [46, 47], by realising the insufficient level of Galilean invariance of the previous MRT operators, introduced the cascaded LBM (CLBM) in which collisions are performed in a reference frame shifted by the macroscopic velocity, where central moments are allowed to relax at different rates in a cascaded manner. Thus, allowing the evolution of higher-order moments to depend on both lower-order moments and hydrodynamic variables, leads to using a discrete equilibrium distribution that contains higher order terms in velocity. In previous SRT and MRT models, the discrete equilibrium distribution was derived under the low Mach, i.e second-order, approximation of the continuous Maxwell-Boltzmann distribution. This approximation is equivalent to a large-wavelength assumption which is incorrect since most instabilities arise from small-wavelength patterns [46, 47]. Furthermore, in order to uniquely determine the equilibrium distribution and achieve Galilean invariance, a given velocity set must be sufficient to adjust different moments independently. Therefore, the common 13, 15 and 19 velocity sets used in three-dimensional LBMs are not applicable in the CLBM formulation.

As a further improvement to the 3D CLBM, Geier *et al.* [48] proposed a novel scheme, where collisions are carried out in the space of cumulants. The use of cumulants is shown to eliminate errors in Galilean invariance and hyper-viscosity while maintaining, or even improve, the stability of the central moment method. Hyper-viscosity in a three-dimensional LBM scheme is defined as $\nu^* = (1/\omega_8 - 1/2)$, where ω_8 is the relaxation rate of the fourth order moments. At relatively short wave lengths, the spurious dissipation from the relaxation of the fourth order moments can be significantly higher than the physical dissipation. Normally, hyper-viscosity is formally two orders smaller in wave number than the shear viscosity and is considered to be asymptotically small. However, as shown by Geier *et al.* [48], at turbulent flows where the kinematic viscosity is very small, the hyper-viscosity can grow to several orders of magnitude higher than the kinematic viscosity and therefore can not be neglected. Arbitrarily lowering the hyper-viscosity would lead to no damping of the non-hydrodynamic observable quantities, introduce spurious dissipation and therefore compromise the numerical stability of the scheme. In the space of cumulants the non-hydrodynamic observable quantities can be dampened without entailing a leading order hyper viscosity [48]. Most recently, Geier *et al.* [191] derived exact functional relationships for the elimination of the linearised leading error of the cumulant lattice Boltzmann method in order to achieve fourth-order accuracy of the diffusion part in the Navier-Stokes equations for small enough viscosity values.

A critical question rises regarding the need of incorporating a turbulent model in the LBM framework in order to investigate high Reynolds number flows both in terms of numerical stability and accuracy. Several studies include incorporation of LES models in the LBM including standard and dynamic Smagorinsky models [166, 167], sigma models [168], wall adaptive models [169] and deconvolution approaches [170]. Most commonly, the implementation of a LES model in the LBM utilises the properties of the LBM and computes the strain rate tensor using only the non-equilibrium parts of the distributions.

The cumulant LBM proposed by Geier *et al.* [48] has shown extremely high stability and accuracy properties at high Reynolds number turbulent flows around a sphere without the need of any stability limiters or turbulent models. However, the need of turbulent models in order to resolve the sub-grid-scales (SGS) has not been discussed. Stiebler *et al.* [166] investigated the flow around a sphere at $Re = 10000$ using a coupled MRT-LBM constant Smagorinsky scheme and shown very good agreement with a NS direct eddy simulation (DES) model. However, the authors used an extremely refined region around the sphere and concluded that the performance and efficiency of the scheme could be improved by incorporating a near-wall compatible SGS model. Similar observations can be found in Premnath *et al.* [167] using a dynamic SGS MRT-LBM to investigate a turbulent channel flow. Finally, the deconvolution model proposed by Malaspinas and Sagaut [170] based on the SRT-LBM has been shown to correctly represent the dynamics of turbulent flows by simulating the turbulent mixing layer.

In order to increase the solution accuracy around the area of interest, whilst maintaining a non-prohibitive computational cost, a grid refinement technique is employed in this study. There exist two major grid refinement approaches in the LBM. In the first one, the flow variables are represented using a volumetric cell-centered method [49, 50, 51], while in the second, a point-wise, cell-vertex approach is used [52, 53]. Filippova and Hanel [52] used an adjustment of the non-equilibrium distributions based on the different relaxation times of each grid level in order to transfer information between grids with different resolution, whilst keeping an equal Reynolds number in all grids. Chen *et al.* [164] presented a grid refinement scheme based on the MRT-LMB and propose additional relationships based on the Chapman-Enskog multiscaling analysis in order to recover the correct hydrodynamic variables. Guzik *et al.* [176] examined the effect of interpolation methods on the accuracy of grid refinement and adaptive grid refinement implementations. Recently, Kuwata and Suga [165] presented a method where a correction step for the macroscopic flow variables is introduced when transferring information across grids with different resolutions in order to remove discontinuities at the interfaces. Lagrava *et al.* [54] proposed a method of the same principle, where a filtering operation was employed when transferring information from fine to coarse grids, where the fine grid scales that can not be resolved by the coarse grid are removed. Following the work of Lagrava *et al.* [54], Falagkaris *et al.* [55, 56] used a different interpolation and filtering schemes and applied the algorithm in the multiple relaxation time cascaded LBM.

The numerical accuracy of a LBM scheme strongly depends on the boundary conditions (BC). The most common question found in the literature is how to replace/reconstruct the missing populations on a boundary node after the streaming step. Translating macroscopic boundary conditions to particle distribution functions is not a trivial task. The first and most popular to date way of prescribing a boundary condition on a flat wall in the LBM framework is the so-called bounce-back rule. In a bounce-back scheme the velocities of particles that hit the walls are reversed. Despite of its simplicity and unconditional stability, the bounce-back scheme needs further modifications in order to be applied to moving boundary applications. In order to improve the low numerical accuracy of the bounce-back rule, many schemes have been proposed to impose Dirichlet velocity boundary conditions.

Chang *et al.* [183] proposed a curved boundary treatment where the distribution functions originating from the solid domain at the boundary nodes are modified using known distribution functions and correctors in order to satisfy the momentum. Zhao and Yong [184] presented a single-node second-order accurate scheme, based on the Maxwell iteration expression of the populations [185], for prescribing Dirichlet BCs on both straight and curved boundaries. Olson [186] modified the well established link bounce back BC in order to decrease discretisation artifacts and sensitivities to the grid alignment. Another modified version of the bounce-back scheme can be found in the work of Yin and Zhang [187]. Verschaeve and Mller [188] developed a curved no-slip boundary condition where the populations are reconstructed from the density, velocity and rate of strain. Boundary treatments for multi speed lattice Boltzmann models can be found in the work of Lee *et al.* [189]. The interested reader should refer to Nash *et al.* [190] for an investigation of three commonly used complex boundary treatments at moderate Reynolds numbers. The interested reader should refer to Zou and He [57], Inamuro [58], Latt and Chopard [59], Skordos [60], Ansumali and Karlin [61], Halliday *et al.* [62] for flat wall treatments and to Bouzidi *et al.* [63], Ginzburg and d'Humières [38], Junk and Yang [26] and Geier *et al.* [48] for curved boundary treatments.

1.2.2 The Immersed Boundary method

One of the challenging and important issues in computational fluid dynamics (CFD) is the accurate and efficient treatment of complex moving or deformable boundaries. Most practical problems involving complex geometries often require unstructured body-conformal grids in order to accurately enforce the boundary conditions. Such approaches involve complicated meshing techniques and increase both the CPU time and the amount of memory used. The solution process can be simplified by developing a non-body-conformal approach where the solution of the governing fluid equations is decoupled from the implementation of the boundary conditions. The immersed boundary method is such an approach and it was initially introduced by Peskin [64] in the 1970s to simulate blood flows in the human heart. In recent years, the IBM has received a great attention in simulating flows with complex geometries[65, 66, 67, 68]. IBM uses a fixed Eulerian grid, usually a Cartesian grid, for the fluid and a Lagrangian set of points, independent of the first, to represent the immersed physical boundary. The boundary is treated as a deformable body with high stiffness, thus a small distortion on the boundary will yield a force that tends to restore the boundary to its original position. The total balanced force is then distributed into the Eulerian grid and the NSE with a body force term are solved for the entire computational domain. Conceptually, the boundary force density in the IBM can be evaluated by either feedback forcing methods or by direct forcing methods.

In the original feedback forcing IBM of Peskin [64] the force is computed from the boundary surface deformation using Hooke's law. The spring constant in the computation can be somehow arbitrarily selected, e.g. a large value has been used in the work of Lai and Peskin [69] in order to simulate the flow around a rigid cylinder. Peskin's method may be considered as a one free parameter feedback forcing IBM, whereas Saiki and Birigen [70] and Goldstein *et al.* [71] introduced additional free parameters in the scheme. The main drawback of feedback forcing

IBM schemes is the tuning of the free parameters which in several occasions, i.e. unsteady flows, may lead to unstable conditions with decreasing numerical accuracy and efficiency. Similar to direct forcing methods, the boundary forces are spread across the boundary.

In the IBM, the immersed physical boundary nodes do not generally coincide with the grid nodes. The Lagrangian and the Eulerian grid sets communicate through an interface scheme. Two communication strategies exist in the IBM framework; the diffuse and the sharp interface schemes. The forcing points in the sharp interface schemes are located on the grid nodes closest to the boundary, i.e. no Lagrangian ensemble exists. Therefore, the forces are not spread across the boundary and the no-slip boundary condition may be satisfied using appropriate velocity interpolation algorithms. In the diffuse scheme the forcing points are placed on the immersed boundary, i.e. Lagrangian nodes. Therefore the no-slip boundary condition is satisfied by distributing the computed forces into the grid. The force distribution is achieved through discrete delta functions.

The sharp interface scheme was used in the first direct forcing IBM by Mohd-Yusof [72] in a spectral frame. Fadlun *et al.* [73] and Kim *et al.* [74] introduced the direct forcing method to finite difference and finite volume solvers respectively. The scheme of Kim *et al.* [74], contrary to the previous methods, removes the arbitrariness of the interpolation direction by placing the force nodes inside the solid domain and using a second-order scheme for the velocity interpolation. In addition, the force is semi-implicitly added to the governing equations, improving the numerical accuracy of the solution. Additional developments in the sharp interface scheme may be found in Balaras [75], Choi *et al.* [76], Iaccarino and Verzicco [77], Ghias *et al.* [78] and Kang and Hassan [79], where either interior or exterior forcing nodes have been selected. The first diffuse interface direct forcing IBM was proposed by Silva *et al.* [80], where both the pressure and the derivatives of the velocity were used in the body force computation. In the work of Uhlmann

[81] the body forces were computed using only the unforced velocities on the neighbouring to the boundary grid nodes. In addition he showed that the diffuse interface schemes removes the spurious oscillations of the sharp forcing methods and provides smoother solutions. Li *et al.* [192] presented an IB-LBM for single and multi-component fluid flows using an improved spreading operator in order to better ensure the no-slip solid boundary condition. An interesting investigation of the boundary slip from the Immersed Boundary Lattice Boltzmann Models can be found in the work of Le and Zhang [123]. The authors reported deviations of the velocity profiles both in the immersed boundary layer and the bulk region. This issued should be further investigated.

The major drawback of the diffuse interface direct forcing IBM schemes is that either because of insufficient interpolation accuracy or the reconstruction of the "unforced" velocity near the boundary, the no slip boundary condition may not be satisfied exactly. To ensure this, several implicit schemes have been developed. In the schemes of Su *et al.* [82] and Le *et al.* [83] the implicitness is enforced by solving large banded matrix equations. Luo *et al.* [84] and Wang *et al.* [85] proposed the multi direct forcing schemes (MDF), where iterative procedures of forcing and force spreading are performed until a given convergence criterion is satisfied. However, as shown by Kang *et al.* [79] the local numerical errors near the boundaries slightly increase as the number of iterations in the MDF increases.

In all IBM, the body forces applied on the boundary and spread into the grid result in movement of the internal to the body fluid. The motion of the internal fluid does not affect the flow characteristics outside the boundary. However, as pointed out by Suzuki and Inamuro [86], if the forces acting on the boundary are obtained by the negative sum of the body forces, as in [69], they are influenced by the motion of the internal mass. There have been only a few studies on the effect of the internal mass on the computation of the aerodynamic forces [87, 81, 88]. Uhlmann [81] treated the internal mass as a rigid body RBA, imposing, however, a limitation on the density ratio of the body to the fluid for a stable simulation. Uhlmann [81] also computed the effect of the internal mass by summing the forces over all internal Eulerian points. The identification of the internal Eulerian points in a moving boundary application is very complex and computationally expensive. There are however efficient methods for the identification of the internal and external to the body Eulerian nodes [75]. A similar observation on the limitation of the density ratio of the body to the fluid has been made by Ladd and Verberg [89]. Feng and Michaelides [88] further developed the work of Uhlmann [81] by eliminating the limitation on the density ratio. Shen *et al.* [90], following the work of Balaras [75], investigated the effect of the internal mass by integrating the NSE with the body forces. The magnitude of the internal mass effects was examined in systems with prescribed boundary motions. Most recently, Suzuki and Inamuro [86] proposed the Lagrangian point approximation (LPA) as an efficient method for describing the internal mass effects and compared it with the previous methods of Uhlmann [81] and Feng and Michaelides [88]. They examined the effect of the internal mass in cases where the body motion is defined by the fluid flow as well as, the dependency of the effect on very low Reynolds number flows.

1.2.3 Coupling

The first step in coupling the LBM and the IBM is the derivation of a suitable discrete force term that will be added to the governing fluid equations. Adding forces to the LBM is not a trivial task and requires careful consideration of the discrete nature of the fluid solver. Luo [91] introduced the force term into the collision term using a similar expression to the one adopted for a previous LGCA model [92]. Shan and Chen [93] incorporated the fluid velocity in the discrete force expression. He *et al.* [34] proposed a term that includes the equilibrium values of the populations. Following the work of He *et al.* [34], Premnath [94] incorporated the forcing terms in the cascaded collision operator. In the work of

Martys *et al.* [95] the force derivative has been evaluated using an expansion in the Hermite polynomial series keeping only the velocity terms up to second order. A similar forcing term has been proposed by Guo *et al.* [96]. Kupershtokh *et al.* [97] proposed the so called exact difference method (EDM), where the discrete forcing term is expressed only in terms of the equilibrium distributions in order to avoid shifting the populations in the velocity space. More recently, Cheng and Li [98] derived a scheme where the effects of both the current and the next time step are incorporated in the source term. Similar to the EDM, the effect of the forcing term is independent of the relaxation parameter and thus can be directly applied to any collision model.

In recent years, many efforts have been made in order to improve the coupling between the IBM and the LBM. In the penalty method, proposed by Feng and Michaelides [99], the immersed boundary is allowed to deform slightly and is restored back to its target position using a linear spring approximation. Dupuis *et al.* [100] presented a direct-forcing IBM, where the boundary force is computed using the interpolated velocity and a desired reference velocity. The momentum exchange of the particle distributions at the boundary was used by Niu *et al.* [101] to calculate the force acting on the immersed boundary. However, the non-slip boundary condition cannot be satisfied exactly by those methods. A few iterative IB schemes exist in the literature [79, 102] that improve the accuracy of the prescribed boundary conditions. Zhang *et al.* [103] proposed an iterative force correction scheme based on Cheng's external forcing term [98]. Similar to the work of Su *et al.* [82] and Le *et al.* [83], Wu and Shu [104] developed an implicit velocity correction-based IB-LBM based on Guo's external forcing term [96]. In the present work, the iterative force correction IB scheme proposed by Zhang *et al.* [103] is used as described in our previous studies [55, 56].

1.3 Summary of the thesis

This thesis is organised as follows. In Chapter 2, the derivation of the cascaded collision operator from the continuous central moment formulation of the Boltzmann equation and a method for incorporating the forces is presented. The basic concepts of grid refinement, as well as the boundary conditions used in this study are explained. In Chapter 3, the discrete formulation of the immersed boundary method is presented focusing on some of the most significant aspects of the method including the interpolation kernels and the discretisation parameters. An overview of the commonly used IBM schemes in the LBM framework is presented. Finally, the effect of the internal mass in the computation of the aerodynamic forces is discussed. In Chapter 4, the coupling between the LBM and the IBM is presented, where emphasis is given on the selection of appropriate discrete forcing terms. In Chapter 5, a novel hybrid higher-order immersed boundary method is described in detail. Focusing on the major drawbacks of the IBM we present two novel strategies. First, the boundary layer is smoothly expanded inside the boundary in order to remove discontinuities in the velocity gradient across the boundary. Second, a pressure correction algorithm is explained. In Chapters 6, 7 and 8, the coupled solver is applied to viscous flows around stationary, moving and wilfully deformed boundaries. Finally, Chapter 9 concludes the thesis.

The thesis presents a novel coupling between the cascaded multi-domain lattice Boltzmann method and an iterative direct forcing immersed boundary method. The resulted numerical scheme is able to accurately and efficiently simulate high Reynolds number flows around moving and wilfully deformable boundaries, as well as to provide great understanding of the complex flow characteristics and aerodynamic loads acting on 2 dimensional bodies. The robustness and numerical accuracy is demonstrated through various industrial applications, where the present numerical results are in very good agreement with experimental and other numerical results. By utilising the mathematical properties of both the fluid and the solid solvers, implementing a multi-domain algorithm and avoiding expensive meshing techniques, the coupled scheme is shown to be very efficient without compromising the overall numerical accuracy. The majority of the presented results are obtained within a few hours of CPU time on a 2.8 GHz Intel Core i7 MacBook Pro computer with a 16GB memory.

Chapter 2

The Lattice-Boltzmann Method

2.1 The Cascaded Lattice-Boltzmann Method

In the first part of this chapter a review of the cascaded lattice-Boltzmann is presented. That includes the theory of the continuous Boltzmann equation and the derivation of the cascaded collision operator. For the sake of completeness, the single relaxation time (SRT) and the multiple relaxation time (MRT) collision models will be discussed. Preliminary results for the flow in a lid driven cavity are presented in order to demonstrate the accuracy and stability of the CLBM at various Reynolds numbers. In the second part of the chapter a discussion on the grid refinement scheme and a brief review on the boundary conditions used in this study is presented. Modifications to the existing schemes in order to increase the accuracy and stability of the methods are discussed in detail.

2.1.1 The Boltzmann equation: Continuous central moment formulation

In the kinetic theory framework, the dynamics of large numbers of particles, interacting through collisions, are of great interest. More emphasis is given on the understanding of the macroscopic state behaviours rather than on individual particle states, as the latter is in principle impossible to achieve. Let us define the particle mass distribution function (PDF) $f(\mathbf{x}, \boldsymbol{\xi}, t)$ such that $f(\mathbf{x}, \boldsymbol{\xi}, t)d^3xd^3\xi$, is the total mass of the particles located around the position $\mathbf{x}$ in the volume d^3x and have a velocity around $\boldsymbol{\xi}$ in the volume $d^3\xi$ at time t. This approximation is only valid when a large number of particles is contained in the small volume element $d^3xd^3\xi$. The zeroth and first order moments of the PDF are defined as

$$\rho(\mathbf{x}, t) = \int d\boldsymbol{\xi}\, f(\mathbf{x}, \boldsymbol{\xi}, t), \tag{2.1}$$

$$\mathbf{j}(\mathbf{x}, t) = \rho(\mathbf{x}, t)\mathbf{u}(\mathbf{x}, t) = \int d\boldsymbol{\xi}\, \boldsymbol{\xi}\, f(\mathbf{x}, \boldsymbol{\xi}, t), \tag{2.2}$$

where ρ is the density and $\mathbf{u}$ is the mean velocity field. The integration is carried out on the whole velocity space. In addition, let us define the microscopic velocity as $\mathbf{c}(\boldsymbol{\xi}, \mathbf{x}, t) = \boldsymbol{\xi} - \mathbf{u}(\mathbf{x}, t)$. Higher order moments are computed with respect to $\mathbf{u}(\mathbf{x}, t)$. The second order moment is computed as

$$\rho\epsilon = \frac{1}{2}\int d\mathbf{c}\, \mathbf{c}^2\, f(\mathbf{x}, \boldsymbol{\xi}, t), \tag{2.3}$$

where ϵ is the internal energy. For mono-atomic gases, the internal energy ϵ is related to the temperature T by the equipartition of energy $\rho\epsilon = D\rho\theta/2$, where $\theta = k_BT/m$, k_B is the Boltzmann constant, m is the mass of the particle and D is the physical dimension.

Let us consider a two-dimensional athermal fluid and let only the density

$\rho(\mathbf{x},t)$ and the velocity $\mathbf{u}(\mathbf{x},t)$, neglecting the presence of any force sources, to characterize its local hydrodynamic behavior at a Cartesian coordinate system (x,y). In a continuous particle velocity space (ξ_x,ξ_y) the local Maxwell-Boltzmann distribution is given by

$$f^{\mathscr{M}} = \frac{\rho(\mathbf{x},t)}{(2\pi\theta(\mathbf{x},t))^{D/2}} \exp\left[-\frac{(\boldsymbol{\xi}-\mathbf{u}(\mathbf{x},t))^2}{2\theta(\mathbf{x},t)}\right] = \frac{\rho(\mathbf{x},t)}{2\pi c_s^2}\exp\left[-\frac{(\boldsymbol{\xi}-\mathbf{u}(\mathbf{x},t))^2}{2c_s^2}\right], \tag{2.4}$$

where $c_s^2 = 1/3$ is the speed of sound for the D2Q9 model. The continuous moments of the Maxwell-Boltzmann distribution function are given by

$$\widehat{\mu}^{\mathscr{M}}_{x^m y^n} = \int\limits_{-\infty}^{\infty}\int\limits_{-\infty}^{\infty} \xi_x^m \xi_y^n f^{\mathscr{M}}(\xi_x,\xi_y)d\xi_x d\xi_y, \tag{2.5}$$

where m,n are integers and $x^m y^n$ means $xxx\cdots m$ and $yyy\cdots n$ - times. The use of "*hat*" will represent a value in moment space unless otherwise specified. Similarly to the definition of the continuous moments (6.8), the central moments of the Maxwell-Boltzmann distribution are given by

$$\widehat{M}^{\mathscr{M}}_{x^m y^n} = \int\limits_{-\infty}^{\infty}\int\limits_{-\infty}^{\infty} f^{\mathscr{M}}(\xi_x-u_x)^m(\xi_y-u_y)^n d\xi_x d\xi_y. \tag{2.6}$$

The central moments for the equilibrium distribution are constant. Based on the fact that $f^{\mathscr{M}}$ is by definition an even function, the odd moments vanish, i.e. $\widehat{M}^{\mathscr{M}}_{x^m y^n} = 0$ when m or n are odd. The even moments $\widehat{M}^{\mathscr{M}}_{x^m y^n}$ are chosen from the Maxwell-Boltzmann distribution for an isothermal ideal gas [46]. Therefore, $\widehat{M}^{\mathscr{M}}_0 = \rho$, $\widehat{M}^{\mathscr{M}}_{xx} = \widehat{M}^{\mathscr{M}}_{yy} = c_s^2\rho$ and $\widehat{M}^{\mathscr{M}}_{xxyy} = c_s^4\rho$.

In the presence of an external force field $\mathbf{F}$ the Boltzmann equation may be written as

$$\Big(\theta_t + \boldsymbol{\xi}\cdot\boldsymbol{\nabla}_x + \mathbf{F}\cdot\boldsymbol{\nabla}_\xi\Big)f(\mathbf{x},\boldsymbol{\xi},t) = \boldsymbol{\Omega}\big(f-f^{\mathscr{M}}\big), \tag{2.7}$$

where $\boldsymbol{\Omega}$ is a general scattering operator and $\mathbf{F}$ is an external body force which can depend on both space and time. Various forms of the external body force term have been proposed in the literature and will be further discussed in Section 2.3. In order to incorporate the external forces in the cascaded collision operator, the ansatz proposed by He *etal*. [34] is used. Noting the dependence of the distribution function on the microscopic velocity which is not known, the derivative $\mathbf{F} \cdot \nabla_{\xi} f$ cannot be calculated directly. He *et al.* assumed that since $f^{\mathscr{M}}$ is the leading part of the distribution f and the gradient of $f^{\mathscr{M}}$ contributes the most on the gradient of f, the following approximation can be made:

$$\mathbf{F} \cdot \boldsymbol{\nabla}_{\xi} f \approx \mathbf{F} \cdot \boldsymbol{\nabla}_{\xi} f^{\mathscr{M}} = -\frac{\boldsymbol{\xi} - \mathbf{u}}{c_s^2} f^{\mathscr{M}} \tag{2.8}$$

Thus, the third term on the LHS of Eq. (2.7) that describes change in f by the presence of a force field is given by the following ansatz:

$$\Delta f^{\mathscr{F}} = \frac{\mathbf{F}}{\rho} \cdot \frac{(\boldsymbol{\xi} - \mathbf{u})}{c_s^2} f^{\mathscr{M}} \tag{2.9}$$

The corresponding continuous central moments of the modified distribution function due to the presence of a force field are defined as

$$\widehat{\Gamma}^{\mathscr{F}}_{x^m y^n} = \int\limits_{-\infty}^{\infty} \int\limits_{-\infty}^{\infty} \Delta f^{\mathscr{F}} (\xi_x - u_x)^m (\xi_y - u_y)^n d\xi_x d\xi_y \tag{2.10}$$

It occurs that, $\widehat{\Gamma}^{\mathscr{F}}_{x} = F_x$, $\widehat{\Gamma}^{\mathscr{F}}_{y} = F_y$, $\widehat{\Gamma}^{\mathscr{F}}_{xxy} = c_s^2 F_y$ and $\widehat{\Gamma}^{\mathscr{F}}_{xyy} = c_s^2 F_x$. All other moments vanish.

2.1.2 Discrete formulation of the Boltzmann Equation

For two-dimensional flows, the nine-velocity square lattice model, often referred to as the 2D 9-velocity (D2Q9) model [11], which is shown in Fig. 2.1 has been

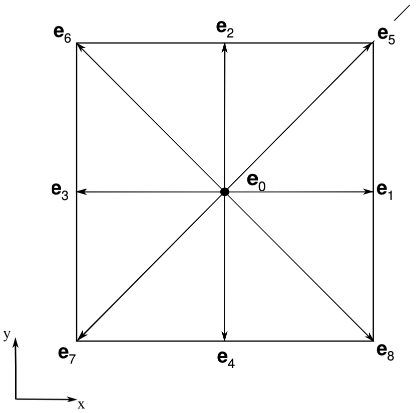


Figure 2.1: Two-dimensional, nine-velocity (D2Q9) Lattice structure

succesfully used in the literature. The discrete velocity set is given by:

$$\mathbf{e}_\alpha = \begin{cases} (0,0), & \alpha = 0 \\ c(\cos[(\alpha-1)\pi/2], \sin[(\alpha-1)\pi/2]), & \alpha = 1-4 \\ \sqrt{2}c(\cos[(2\alpha-9)\pi/4], \sin[(2\alpha-9)\pi/4]), & \alpha = 5-8 \end{cases} \tag{2.11}$$

Here and henceforth, Latin and Greek subscripts are used for the Cartesian coordinate and particle velocity directions, respectively. In this paper, a row vector of any state variable χ along a for $a = 0, \ldots, 8$ will be represented by the "*bra*" operator $\langle \chi |$, i.e. $(\chi_0, \chi_1, \chi_2, \ldots, \chi_8)$ whereas, a column vector will be represented by the "*ket*" operator $|\chi\rangle$, i.e. $(\chi_0, \chi_1, \chi_2, \ldots, \chi_8)^\dagger$, where the superscript † is the transpose operator. Accordingly, the inner-product is denoted by $\langle \chi | x \rangle$.

Starting from Eq. (2.7), let us define a discrete distribution function for the discrete velocity set $\mathbf{e}_a$ of Eq. (2.11) as

$$\mathbf{f} = |f_a\rangle = (f_0, f_1, f_2, \ldots, f_8)^\dagger. \tag{2.12}$$

In a discrete velocity space the moments are given by

$$\widehat{m}_{x^m y^n} = \sum_{\alpha=0}^{i} e_{\alpha x}^m e_{\alpha y}^n f_\alpha = \big\langle e_{\alpha x}^m e_{\alpha y}^n | f_\alpha \big\rangle. \tag{2.13}$$

The respective discrete central moments of the distribution function are given by

$$\hat{\kappa}_{x^m y^n} = \sum_{\alpha} f_\alpha (e_{\alpha x} - u_x)^m (e_{\alpha y} - u_y)^n = \big\langle (e_{\alpha x} - u_x)^m (e_{\alpha y} - u_y)^n | f_\alpha \big\rangle \tag{2.14}$$

Collision will be represented as a cascaded process in which the higher order moments are influenced by the collision effects on the lower order moments. Thus, the collision can be represented using Geier's [46, 47] notation as

$$\mathbf{\Omega}_a = (\mathbf{K} \cdot \hat{\mathbf{k}})_a, \tag{2.15}$$

where the changes in the discrete moment space are determined by

$$\hat{\mathbf{k}} = |\hat{k}_a\big\rangle = \big(\hat{k}_0, \hat{k}_1, \hat{k}_2, \ldots, \hat{k}_8)^\dagger. \tag{2.16}$$

In the CLBM the Gallilean invariance is imposed on the moments by mapping the configuration space to an equivalent moment space as shown in [46, 47]. Therefore, constraints such as invariance under translation, rotation and collisions must be fulfilled. Following the process presented by Geier[46], the orthogonal set of the basis vectors can be explicitly written as

$$\mathbf{K} = \begin{pmatrix} 1 & 0 & 0 & -4 & 0 & 0 & 0 & 0 & 4 \\ 1 & 1 & 0 & -1 & -1 & 0 & 0 & 2 & -2 \\ 1 & 0 & 1 & -1 & 1 & 0 & 2 & 0 & -2 \\ 1 & -1 & 0 & -1 & -1 & 0 & 0 & -2 & -2 \\ 1 & 0 & -1 & -1 & 1 & 0 & -2 & 0 & -2 \\ 1 & 1 & 1 & 2 & -0 & 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & 2 & 0 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 2 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & 2 & 0 & -1 & 1 & -1 & 1 \end{pmatrix} \tag{2.17}$$

Note that $KK^\dagger$ is a diagonal matrix. It is noteworthy to compare the transofrmation matrix $\mathbf{K}$ with the one considered by Lallemand and Luo [37] for the MRT scattering operator. Here, the ordering of the orthogonal basis vectors $|K_i\rangle$ is based on the ascending powers of moments, whereas in [37] it is based on the increasing powers of the tensorial order of the moments. For more information on the construction of matrix $\mathbf{K}$ the reader is encouraged to refer to [46] and [94].

The forcing term is discretized in the same manner The changes in $\mathbf{f}$ due to the presence of an external force field are defined by the discrete source term S_α as

$$\mathbf{S} = |S_\alpha\rangle = \big(S_0, S_1, S_2, \ldots, S_8\big)^\dagger. \tag{2.18}$$

2.1.3 The Single-Relaxation-Time Lattice Boltzmann Method

In the single-relaxation-time (SRT) Lattice Boltzmann Method, collision are performed in the discrete velocity space $\mathbf{e}_\alpha$ as

$$\bar{f}_\alpha^p = \bar{f}_\alpha - \frac{1}{\tau}(f_\alpha - f_\alpha^{eq}) + S_\alpha, \tag{2.19}$$

where $S_\alpha = (1 - 1/2\tau)F_\alpha$. The relaxation time is defined as $\tau = 1/\omega$, where ω is the relaxation rate. For the D2Q9 model the relaxation rate relates to the shear kinematic viscosity as

$$\nu = c_s^2\left(\frac{1}{\omega} - \frac{1}{2}\right), \tag{2.20}$$

where $c_s = 1/\sqrt{3}$ is the speed of sound for the D2Q9 model. In this simple formulation, all modes are allowed to relax at the same time τ during the collision process [31]. For a detailed expression of the source term S_α in velocity space, see, for example, Guo *et al.* [96]. The discrete equilibrium distribution function is expressed as $f_\alpha^{eq} = w_\alpha \rho\big\{1 + (\mathbf{e}\cdot\mathbf{u})/c_s^2 + (\mathbf{e}\cdot\mathbf{u})^2/2c_s^4 - (\mathbf{u}\cdot\mathbf{u})/2c_s^2\big\}$

2.1.4 The Multiple-Relaxation-Time Lattice Boltzmann Method

The Multiple-Relaxation-Time (MRT) Lattice-Boltzmann Method will be briefly discussed in this section. In the MRT-LBM formulation, both the collision and the forcing steps are performed in the raw moment space. Let us define the equilibrium function and the source terms in velocity space as $\mathbf{f}^{eq} = |f_a^{eq}\rangle = \big(f_0^{eq}, f_1^{eq}, f_2^{eq}, \ldots, f_8^{eq}\big)^\dagger$ and $\mathbf{S} = |S_a\rangle = \big(S_0, S_1, S_2, \ldots, S_8\big)^\dagger$ respectively. The post-collision states of the distribution functions are obtained as $\bar{f}_\alpha^p = \bar{f}_\alpha + \bar{\boldsymbol{\omega}}(\mathbf{x}, t)$, where

$$\bar{\boldsymbol{\omega}}(\mathbf{x},t) = \mathbf{T}^{-1}\Big[-\hat{\boldsymbol{\Lambda}}\big(\hat{\mathbf{f}}-\mathbf{f}^{\widehat{eq}}\big)+\Big(\mathbf{I}-\frac{1}{2}\hat{\boldsymbol{\Lambda}}\Big)\hat{\mathbf{S}}\Big]. \tag{2.21}$$

Here, $\hat{\mathbf{f}}$, $\mathbf{f}^{\widehat{eq}}$ and $\hat{\mathbf{S}}$ are the raw moments of the distribution function, its equilibrium and source terms, defined as $\hat{\mathbf{f}} = \mathbf{T}\mathbf{f}$, $\mathbf{f}^{\widehat{eq}} = \mathbf{T}\mathbf{f}^{eq}$ and $\hat{\mathbf{S}} = \mathbf{T}\mathbf{S}$, where $\mathbf{T}$ is the transformation matrix [37]. The relaxation time matrix is defined as $\hat{\boldsymbol{\Lambda}} = diag(\omega_0, \omega_1, \omega_2, \ldots, \omega_8)$, in which ω_7 and ω_8 determine the shear kinematic viscosity as in Eq. (2.20) and the rest can be tuned to improve numerical stability. For a detailed description of the source term $\hat{\mathbf{S}}$ in raw moment space, see [105]. Finally, the moments of the equilibrium distribution may be written compactly as: $f_0^{\widehat{eq}} = \rho$, $f_1^{\widehat{eq}} = -2\rho + 3\mathbf{j}\cdot\mathbf{j}/\rho$, $f_2^{\widehat{eq}} = \rho - 3\mathbf{j}\cdot\mathbf{j}/\rho$, $f_3^{\widehat{eq}} = j_x$, $f_4^{\widehat{eq}} = -j_x$, $f_5^{\widehat{eq}} = j_y$, $f_6^{\widehat{eq}} = -j_y$, $f_7^{\widehat{eq}} = (j_x^2 - j_y^2)/\rho$ and $f_8^{\widehat{eq}} = (j_x j_y)/\rho$, where $\mathbf{j} = (j_x, j_y) = (\rho u_x, \rho u_y)$ is the momentum vector.

2.1.5 Construction of the cascaded collision operator

The construction of the cascaded collision operator and forcing terms is similar to the one presented in [94] using the orthogonalized set of basis vectors proposed by Geier. The form of both the discrete forcing term S_a and the discrete moments $\hat{k}_a$ is not yet known. Using the definitions in Eqns. (2.12) - (2.18), the discrete evolution equation can be written as

$$f_\alpha(\mathbf{x}+\mathbf{e}_\alpha\Delta t, t+\Delta t) = f_\alpha(\mathbf{x},t) + \Omega_a(\mathbf{x},t) + \int_t^{t+dt} \frac{\mathbf{F}}{\rho}\cdot\frac{(\mathbf{e}_a-\mathbf{u})}{c_s^2} f^{\mathscr{M}} dt. \tag{2.22}$$

The integral on the RHS of Eq. (2.22) represents the cumulative effect of the forces on the discrete distribution functions, thus the particle populations are continuously affected by this as they advect along the discrete directions shown in fig. 2.1 [94]. A second-order trapezoidal rule is used for the calculation of the integral over$t, t + \Delta t$ [94, 34].

$$\int_{t}^{t+dt} \frac{\mathbf{F}}{\rho} \cdot \frac{(\mathbf{e}_\alpha - \mathbf{u})}{c_s^2} f^{\mathscr{M}} dt = \frac{1}{2}\big[S_{\alpha(\mathbf{x},t)} + S_{\alpha(\mathbf{x}+\mathbf{e}_\alpha \Delta t, t+\Delta t)}\big]. \tag{2.23}$$

The implicitness of Eq. (2.23) is removed by transforming the distribution functions along the characteristics according to

$$\bar{f}_\alpha = f_\alpha - 0.5 S_\alpha. \tag{2.24}$$

Using (2.23) - (2.24), the lattice Boltzmann equation, Eq. (2.22) becomes

$$\bar{f}_\alpha(\mathbf{x} + \mathbf{e}_\alpha \Delta t, t + \Delta t) = \bar{f}_\alpha(\mathbf{x}, t) + \Omega_a(\mathbf{x}, t) + S_{\alpha(\mathbf{x},t)}. \tag{2.25}$$

Both the distribution functions and the transformed distribution functions, as well as the source terms are used to define their respective discrete central moments

$$\hat{\kappa}_{x^m y^n} = \sum_\alpha f_\alpha (e_{\alpha x} - u_x)^m (e_{\alpha y} - u_y)^n = \big\langle (e_{\alpha x} - u_x)^m (e_{\alpha y} - u_y)^n | f_\alpha \big\rangle, \tag{2.26a}$$

$$\hat{\bar{\kappa}}_{x^m y^n} = \sum_\alpha \bar{f}_\alpha (e_{\alpha x} - u_x)^m (e_{\alpha y} - u_y)^n = \big\langle (e_{\alpha x} - u_x)^m (e_{\alpha y} - u_y)^n | \bar{f}_\alpha \big\rangle, \tag{2.26b}$$

$$\hat{\sigma}_{x^m y^n} = \sum_\alpha S_\alpha (e_{\alpha x} - u_x)^m (e_{\alpha y} - u_y)^n = \big\langle (e_{\alpha x} - u_x)^m (e_{\alpha y} - u_y)^n | S_\alpha \big\rangle. \tag{2.26c}$$

In addition, Eq. (2.26a) is also valid for the equilibrium state of the distributions f_α and $\bar{f}_\alpha$. Using the continuous central moments formulation of Eqns. (2.6) and (2.10) and by equating them with the respective discrete central moments for the equilibrium state of both distribution functions and source terms we obtain $\widehat{M}^{\mathscr{M}}_{x^m y^n} = \hat{\kappa}^{eq}_{x^m y^n}$ and $\widehat{\Gamma}^{\mathscr{F}}_{x^m y^n} = \hat{\sigma}_{x^m y^n}$. Using Eq. (2.24) and the discrete central moments for the equilibrium states, it follows that

$$\hat{\bar{\kappa}}^{eq}_{x^m y^n} = \hat{\kappa}_{x^m y^n} - 0.5 \hat{\sigma}_{x^m y^n}. \tag{2.27}$$

Eq. (2.27) can be written explicitly as

$$
\begin{aligned}
\hat{\kappa}^{eq}_{0} &= \rho, && (2.28a)\\
\hat{\kappa}^{eq}_{x} &= -0.5F_x, && (2.28b)\\
\hat{\kappa}^{eq}_{y} &= -0.5F_y, && (2.28c)\\
\hat{\kappa}^{eq}_{xx} &= c_s^2\rho, && (2.28d)\\
\hat{\kappa}^{eq}_{yy} &= c_s^2\rho, && (2.28e)\\
\hat{\kappa}^{eq}_{xy} &= 0, && (2.28f)\\
\hat{\kappa}^{eq}_{xxy} &= -0.5c_s^2F_y, && (2.28g)\\
\hat{\kappa}^{eq}_{xyy} &= -0.5c_s^2F_x, && (2.28h)\\
\hat{\kappa}^{eq}_{xxyy} &= c_s^4\rho. && (2.28i)
\end{aligned}
$$

The next step includes the derivation of the raw moments from the central moments using the binomial theorem for any state variable ϕ "*living*" on the discrete velocity set $\mathbf{e}_\alpha$ [94]. Using the compact notation $C^p_q = \frac{p!}{(q!(p-q)!)}$, the central and the raw moments are related through

$$
\begin{aligned}
\big\langle (e_{\alpha x}-u_x)^m(e_{\alpha y}-u_y)^n|\phi\big\rangle = {}& \big\langle e^m_{\alpha x}e^n_{\alpha y}|\phi\big\rangle + \big\langle e^m_{\alpha x}\left[\sum_{j=1}^{n} C^n_j e^{n-j}_{\alpha y}(-1)^j u^j_y\right]|\phi\big\rangle\\
&+ \big\langle e^n_{\alpha y}\left[\sum_{i=1}^{m} C^m_i e^{m-i}_{\alpha x}(-1)^i u^i_x\right]|\phi\big\rangle\\
&+ \big\langle \left[\sum_{i=1}^{m} C^m_i e^{m-i}_{\alpha x}(-1)^i u^i_x\right]\left[\sum_{j=1}^{n} C^n_j e^{n-j}_{\alpha y}(-1)^j u^j_y\right]|\phi\big\rangle.
\end{aligned}
\tag{2.29}
$$

This specific form of S_α is ontained by applying the binomial theorem (2.29) on Eq. (2.26c) and employing the orthogonality properties of $\mathbf{K}$. The discrete force terms in Premnath's method can be explicitly written as

$$S_0 = -2F_xu_x - 2F_yu_y + 2F_xu_xu_y^2, \tag{2.30a}$$

$$S_1 = 1/2F_x + F_xu_x - 1/2F_xu_y^2 - F_yu_yu_x - F_xu_xu_y^2 - F_yu_yu_x^2, \tag{2.30b}$$

$$S_2 = 1/2F_y + F_y u_y - 1/2F_y u_x^2 - F_x u_x u_y - F_x u_x u_y^2 - F_y u_y u_x^2, \tag{2.30c}$$

$$S_3 = -1/2F_x + F_x u_x + 1/2F_x u_y^2 + F_y u_y u_x - F_x u_x u_y^2 - F_y u_y u_x^2, \tag{2.30d}$$

$$S_4 = -1/2F_y + F_y u_y + 1/2F_y u_x^2 + F_x u_x u_y - F_x u_x u_y^2 - F_y u_y u_x^2, \tag{2.30e}$$

$$\begin{aligned} S_5 = {} & 1/4F_x u_y + 1/4F_y u_x + 1/4F_x u_y^2 + 1/4F_y u_x^2 + 1/2F_x u_x u_y \\ & + 1/2F_y u_y u_x + 1/2F_x u_x u_y^2 + 1/2F_y u_y u_x^2, \end{aligned} \tag{2.30f}$$

$$\begin{aligned} S_6 = {} & -1/4F_x u_y - 1/4F_y u_x - 1/4F_x u_y^2 + 1/4F_y u_x^2 + 1/2F_x u_x u_y \\ & - 1/2F_y u_y u_x + 1/2F_x u_x u_y^2 + 1/2F_y u_y u_x^2, \end{aligned} \tag{2.30g}$$

$$\begin{aligned} S_7 = {} & 1/4F_x u_y + 1/4F_y u_x - 1/4F_x u_y^2 - 1/4F_y u_x^2 - 1/2F_x u_x u_y \\ & - 1/2F_y u_y u_x + 1/2F_x u_x u_y^2 + 1/2F_y u_y u_x^2, \end{aligned} \tag{2.30h}$$

$$\begin{aligned} S_8 = {} & -1/4F_x u_y - 1/4F_y u_x + 1/4F_x u_y^2 - 1/4F_y u_x^2 - 1/2F_x u_x u_y \\ & + 1/2F_y u_y u_x + 1/2F_x u_x u_y^2 + 1/2F_y u_y u_x^2. \end{aligned} \tag{2.30i}$$

In order to be conceptually consistent, the non-zero contribution of the higher order central moments, i.e. $\widehat{\Gamma}^{\mathscr{F}}_{x^m y^n} = 0$ for $m+n > 1$, that appears even when the fluid is at rest under homogeneous forces, has been eliminated.

The hydrodynamic variables are obtained by taking the zeroth and first moments of the distributions as

$$\rho = \sum_{\alpha}^{q} \bar{f}_\alpha = \langle \bar{f}_\alpha | \rho \rangle \tag{2.31}$$

$$\rho u_i = \sum_{\alpha}^{q} \bar{f}_\alpha e_{\alpha i} + \frac{1}{2} F_i = \langle \bar{f}_\alpha | e_{\alpha i} \rangle + \frac{1}{2} F_i, \quad i \in x, y. \tag{2.32}$$

The construction of the scattering operation in the cascaded model is initially based on the assumption that the post collision state is the equilibrium state. Once constructed, this assumption is discarded and the moments are allowed to relax during collision. Since mass and momentum are collisional invariants, they are not affected by the collision process. The derivation is the same as the one presented in [46, 47] with the difference that here the forces are included in the collision and thus, should be incorporated in the scattering operator. The complete derivation of the collision operator can be found in [94]. Starting from the lowest order non-conservative equilibrium moments, Eqns. (2.28d) - (2.28i), and using the binomial theorem in Eq. (2.29), as well as the raw moments of the collision kernel $\sum_\alpha (K \cdot \hat{\mathbf{k}})_\alpha e^m_{\alpha x} e^n_{\alpha y} = \sum_\beta \left\langle K_\beta | e^m_{\alpha x} e^n_{\alpha y} \right\rangle \hat{k}_\beta$ we get

$$\begin{aligned}
\hat{k}_3 &= \omega_3 \frac{1}{12} \Big\{ -(\bar{f}_1 + \bar{f}_2 + \bar{f}_3 + \bar{f}_4 + 2(\bar{f}_5 + \bar{f}_6 + \bar{f}_7 + \bar{f}_8)) + \frac{2}{3}\rho \\
&\qquad + \rho(u_x^2 + u_y^2) - (F_x u_x + F_y u_y) \Big\}, && (2.33) \\
\hat{k}_4 &= \omega_4 \frac{1}{4} \Big\{ (\bar{f}_2 + \bar{f}_4 - \bar{f}_1 - \bar{f}_3) + \rho(u_x^2 - u_y^2) - (F_x u_x - F_y u_y) \Big\}, && (2.34) \\
\hat{k}_5 &= \omega_5 \frac{1}{4} \Big\{ (\bar{f}_5 + \bar{f}_7 - \bar{f}_6 - \bar{f}_8) - \rho u_x u_y + \frac{1}{2}(F_x u_y + F_y u_x) \Big\}. && (2.35)
\end{aligned}$$

The first three non-conservative moments are independent of each other thus, they only depend on the macroscopic variables and the distribution functions. Therefore, since $\hat{k}_\alpha \equiv \hat{k}_\alpha(f)$ for $\alpha = 3, 4, 5$, the relaxation rate is applied to the whole moment. For higher-order moments, $\hat{k}_\alpha \equiv \hat{k}_\alpha(f, \hat{k}_\beta)$ for $\beta = 0, 1, 2, \ldots, \alpha - 1$, that depend on lower-order post collision states, only the independent to those states parts are relaxed.

$$\begin{aligned}
\hat{k}_6 &= \omega_6 \frac{1}{4} \Big\{ (\bar{f}_5 + \bar{f}_6 - \bar{f}_7 - \bar{f}_8 - 2u_x(\bar{f}_5 + \bar{f}_7 - \bar{f}_6 - \bar{f}_8) \\
&\quad - u_y(\bar{f}_1 + \bar{f}_3 + \bar{f}_5 + \bar{f}_6 + \bar{f}_7 + \bar{f}_8)) + 2\rho u_x^2 u_y \\
&\quad + \frac{1}{2}(1 - u_x^2) F_y - F_x u_x u_y \Big\} \\
&\quad - 2u_x \hat{k}_5 - \frac{1}{2} u_y (3\hat{k}_3 + \hat{k}_4), && (2.36) \\
\hat{k}_7 &= \omega_7 \frac{1}{4} \Big\{ (\bar{f}_5 + \bar{f}_8 - \bar{f}_6 - \bar{f}_7 - 2u_y(\bar{f}_5 + \bar{f}_7 - \bar{f}_6 - \bar{f}_8) \\
&\quad - u_x(\bar{f}_2 + \bar{f}_4 + \bar{f}_5 + \bar{f}_6 + \bar{f}_7 + \bar{f}_8)) + 2\rho u_x u_y^2 \\
&\quad + \frac{1}{2}(1 - u_y^2) F_x - F_y u_x u_y \Big\} \\
&\quad - 2u_y \hat{k}_5 - \frac{1}{2} u_y (3\hat{k}_3 + \hat{k}_4), && (2.37) \\
\hat{k}_8 &= \omega_8 \frac{1}{4} \Big\{ -(\bar{f}_5 + \bar{f}_6 + \bar{f}_7 + \bar{f}_8 - 2u_x(\bar{f}_5 + \bar{f}_8 - \bar{f}_6 - \bar{f}_7)
\end{aligned}$$

$$\begin{aligned}
&- 2u_y(\bar{f}_5 + \bar{f}_6 - \bar{f}_7 - \bar{f}_8) + u_x^2(\bar{f}_2 + \bar{f}_4 + \bar{f}_5 + \bar{f}_6 + \bar{f}_7 \\
&+ \bar{f}_8) + u_y^2(\bar{f}_1 + \bar{f}_3 + \bar{f}_5 + \bar{f}_6 + \bar{f}_7 + \bar{f}_8) + 4u_x u_y(\bar{f}_5 \\
&+ \bar{f}_7 - \bar{f}_6 - \bar{f}_8)) + \frac{1}{9}\rho + 3\rho u_x^2 u_y^2 - (F_x u_x u_y^2 + F_y u_y u_x^2)\Big\} \\
&- 2\hat{k}_3 - \frac{1}{2}u_x^2(3\hat{k}_3 - \hat{k}_4) - \frac{1}{2}u_y^2(3\hat{k}_3 + \hat{k}_4) \\
&- 4u_x u_y \hat{k}_5 - 2u_y \hat{k}_6 - 2u_x \hat{k}_7. \qquad (2.38)
\end{aligned}$$

The post-collision states of the distribution functions, denoted by the superscript p, are obtained as

$$\bar{f}_\alpha^p = \bar{f}_\alpha + (K \cdot \hat{\mathbf{k}})_\alpha + S_\alpha. \qquad (2.39)$$

After the post-collision states $\bar{f}_\alpha^p$ have been computed, the streaming step is performed in the velocity space $\mathbf{e}_\alpha$ as

$$\bar{f}_\alpha(\mathbf{x} + \mathbf{e}_\alpha \Delta t, t + \Delta t) = \bar{f}_\alpha^p(\mathbf{x}, t). \qquad (2.40)$$

A Chapman-Enskog multi scale analysis shows that the kinematic shear viscosity is determined by the relaxation parameters ω_4 and ω_5. The constraint $\omega_4 = \omega_5$ occurs from the requirement of maintaining an isotropic stress tensor. Since the kinematic shear viscosity is a function of the speed of sound and the mean free flight time between two binary collisions, it is given by

$$\nu = c_s^2\left(\frac{1}{\omega_{(4,5)}} - \frac{1}{2}\right). \qquad (2.41)$$

The bulk viscosity is determined by the relaxation parameter ω_3. Geier [46], performed various numerical simulations in order to, find consistent relaxation rates to match higher order moments to physical quantities. A stable numerical approach is to equilibrate higher - order moments, i.e. taking $\omega_6 = \omega_7 = \omega_8 = 1$.

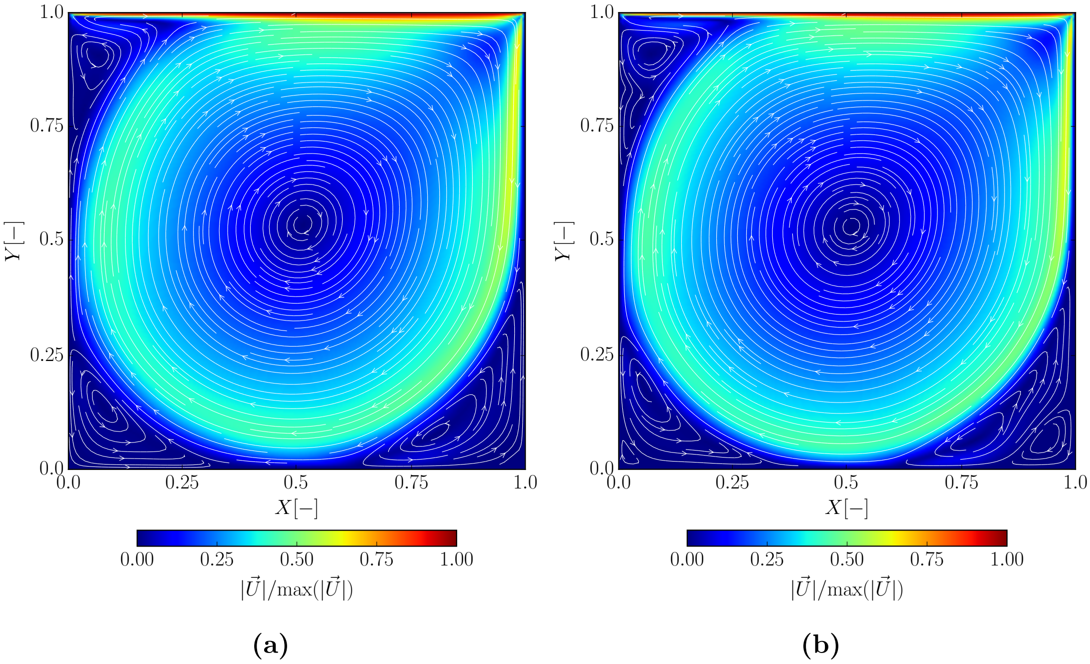


Figure 2.2: Normalised velocity field at (a) $Re = 5000$ and (b) $Re = 10000$.

2.1.6 Test case: 2D Lid driven cavity flow

Figures 2.2 and 2.3 present some initial validation results of the cascaded LBM for the extensively studied 2D lid driven cavity flow. The computational grid consists of 256×256 points. The flow field is initialised with $\rho_0 = 1$ and $U_0 = 0.02$ on the top boundary. The regularised boundary condition as described in 2.5.3 is used for the top boundary and simple bounce back BCs are used for the left, right and bottom boundaries. The results for a range of Reynolds numbers are compared with the computational data of Ghia *et al.* [1] and very good agreement is observed.

2.2 Grid-refinement in the LBM framework

In order to increase the solution accuracy around the area of interest, whilst maintaining a non-prohibitive computational cost, a grid refinement technique

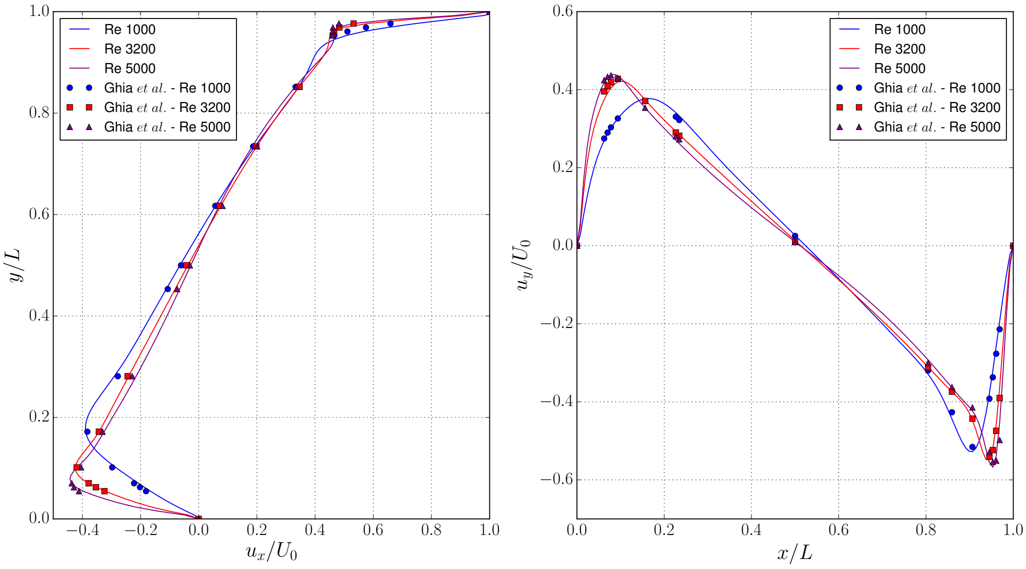


Figure 2.3: Velocity profiles at (a) $x/L = 0.5$ and (b) $y/L = 0.5$ for $Re = 1000, 3200, 5000$. The results are compared with the computational data of Ghia *et al.* [1].

is employed in this study. In this section, the basic concepts of grid refinement in the LBM framework will be discussed. That includes the two different grid refinement techniques used in the literature, the multi-grid and the multi-domain algorithms. Finally, the strategy used in this study will be presented.

Many engineering problems with realistic flows, including a wide range of scales to be resolved, often require excessively high computational power. The most common approach used in the literature employs a localization of the smallest resolvable scales in a limited computational area using locally refined grid algorithm that can spare significant amounts of computational power. The conformal mesh structure of LBM solvers often induces strong discontinuities of the physical quantities at the grid transition regions under a grid refinement operation.

There exist two major grid refinement approaches in the LBM. In the first one, the flow variables are represented using a volumetric cell-centered method [49, 50, 51], while in the second, a point-wise, cell-vertex approach is used [52, 53]. Filippova and Hanel [52] used an adjustment of the non-equilibrium distributions based on the different relaxation times of each grid level in order to transfer information between grids with different resolution, whilst keeping an equal Reynolds number in all grids. Lagrava *et al.* [54] proposed a method of the same principle, where a filtering operation was employed when transferring information from fine to coarse grids, where the fine grid scales that can not be resolved by the coarse grid are removed. In this work, a cell-vertex approach similar to the one of Lagrava *et al.* [54] is considered. The significance of the filtering operation in the decimation process and the interpolation accuracy in the reconstruction process will be discussed.

2.2.1 Basic concepts of grid refinement

In this section the basic concepts of the two major grid refinement techniques will be discussed. In the cell-centered method (multi-grid), the computational domain is discretized with a coarse grid (level 1). At the refinement regions, the finer grid levels overlap the coarse grid, as shown in Fig. 2.4. In the cell-vertex approach, the regions where finer grids are located are taken off the coarse grid, as shown in Fig. 2.5. That leads to higher memory savings as the degrees of freedom (DoF) are drastically reduced. However, the coupling between the grids and the implementation of the algorithm are more complicated. The communication between the grids is done on the boundaries connecting the grids. As shown if Fig. 2.6, the two-way coupling includes exchange of information from coarse to fine and from fine to coarse grids. In this study, the overlapping grid region is two coarse cell width in order to allow for a soother information exchange between the grids. Le us describe the ensemble of coarse points with C and the ensemble of fine points with F. Let use define $\mathbf{x}_{f\to c}$ the fine grid points that are contained in both

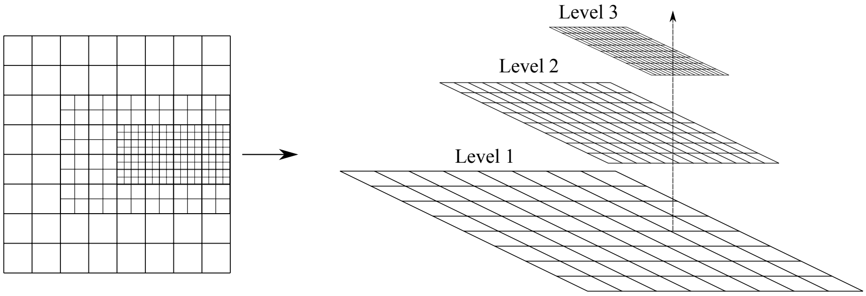


Figure 2.4: Schematic view of the multi-grid approach.

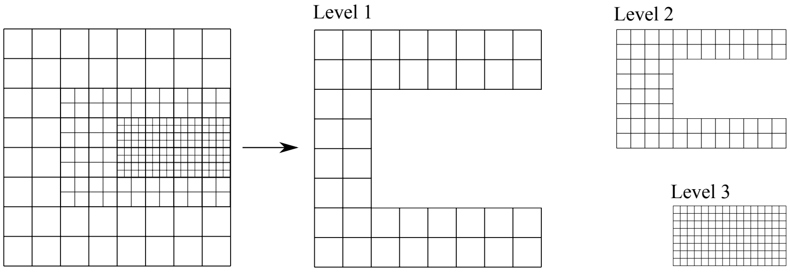


Figure 2.5: Schematic view of the multi-domain approach.

F and C at the boundaries of the coarse grid and $\mathbf{x}_{c\to f}$ the coarse grid points that are contained in both C and F at the boundaries of the fine grid. In addition, let us define $\mathbf{x}^{f}_{c\to f} = \{\mathbf{x} \mid \mathbf{x} \in \mathbf{x}_{c\to f} \text{ and } \mathbf{x} \notin \mathbf{x}^{c}_{c\to f}\}$, $\mathbf{x}^{c}_{c\to f} = \{\mathbf{x} \mid \mathbf{x} \in \mathbf{x}_{c\to f} \text{ and } \mathbf{x} \notin F\}$ and $\mathbf{x}^{c}_{f\to c} = \{\mathbf{x} \mid \mathbf{x} \in \mathbf{x}_{f\to c} \text{ and } \mathbf{x} \notin F\}$.

On the boundaries of the coarse and fine grids, some populations f_α will be missing and need to be reconstructed after a "collide-stream" operation. After a coarse grid "collide-stream" step there are missing information on the boundary nodes of the grid. The same reasoning can be applied to the fine grid, as shown in Fig. 2.7. It is therefore necessary to have a sufficient overlapping region between the two grinds (see Fig. 2.6) in order to be able to reconstruct the missing information.

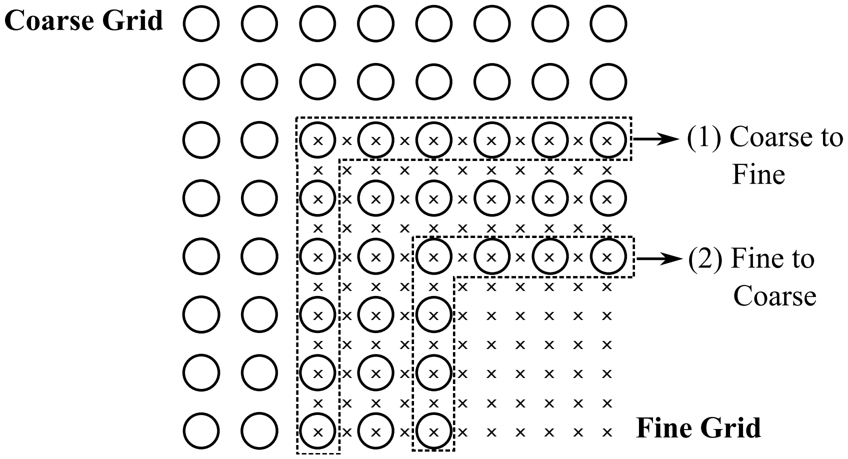


Figure 2.6: Schematic of the places where the copies are performed.

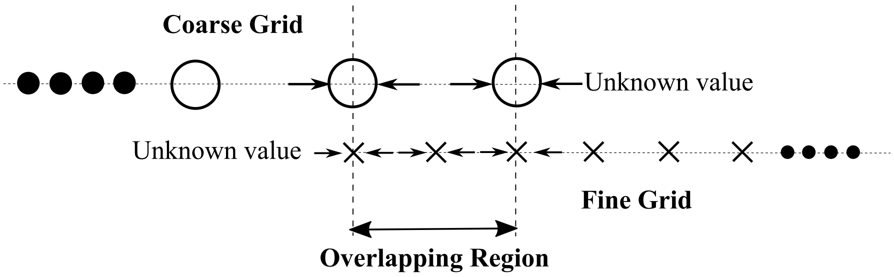


Figure 2.7: Overlapping zone between coarse and fine grids.

2.2.2 Rescaling of physical quantities

When information is transferred between grids of different resolution, the physical quantities must be rescaled. Let us define the spatial resolution of the coarse and fine grids as δx_c and δx_f respectively. In this study, consecutive grids are refined by a factor two, thus $\delta x_f = \delta x_c/2$. In order to achieve higher numerical efficiency, the convective scaling is used for the temporal grid refinement. A detailed discussion on the effects of the scaling schemes in the accuracy of the LBM can be found in Appendix A.1. In the convective scheme the temporal scale is proportional to the spatial scale ($\delta t \sim \delta x$) leading to $\delta t_f/\delta x_f = \delta t_c/\delta x_c =$ constant.

Therefore, two "collide-stream" operations must be performed on the fine grid for every one "collide-stream" operation on the coarse grid. In terms of the physical quantities, both the lattice velocity and pressure are continuous fields on the grid transition region and only the lattice kinematic viscosity ν must be rescaled. Let us define a characteristic length scale L_m, a characteristic velocity U_m and a characteristic kinematic viscosity ν_m, where the subscript m refers to either c or f. In terms of the physical characteristic velocity U and length scale L we get $U_m = U\delta t_m/\delta x_m$ and $L_m = L/\delta x_m$. The Reynolds number is defined as $Re_m = U_m L_m/\nu_m$. Imposing that Reynolds number is independent of the grid refinement level it is shown that $Re_c = Re_f \Leftrightarrow \frac{UL\delta t_c}{\delta x_c^2 \nu_c} = \frac{UL\delta t_f}{\delta x_f^2 \nu_f}$. The kinematic lattice viscosity may then be rescaled as $\nu_f = \frac{\delta x_c}{\delta x_f}\nu_c$. Finally, using Eq. 2.41 the relaxation parameters ω_4 and ω_5 can be rescaled as

$$\omega_f = \frac{2\delta x_f \omega_c}{\delta x_f \omega_c + 2\delta x_c - \delta x_c \omega_c} = \frac{2\omega_c}{4 - \omega_c}. \tag{2.42}$$

A more detailed description of the conversion strategy can be found in Appendix A.2. The consistency and accuracy of the present scheme at the interface of grids with different resolution levels is demonstrated by investigation of the flow around a stationary cylinder. It is shown that both the pressure and the shear stresses are continuous quantities across the interface.

2.2.3 Two-way coupling

Following the work of Dupuis [53] and Lagrava *et al.* [54] the rescaling of the distribution functions is performed as follows. In the LBM, the distribution functions can be written as a summation of the equilibrium and non-equilibrium parts as $f_{\alpha,m} = f_\alpha^{eq}(\rho_m, \mathbf{u}_m) + f_{\alpha,m}^{neq}(\nabla\mathbf{u}) = f_\alpha^{eq}(\rho, \mathbf{u}) + f_{\alpha,m}^{neq}(\nabla\mathbf{u})$, where $\mathbf{u}_f = \mathbf{u}_c = \mathbf{u}$ and $\rho_f = \rho_c = \rho$ since both fields are continuous and independent of the grid resolution.

Finally, in order to allow for a continuous transition of the physical quantities at the grid interface, the reconstruction of the fine and coarse distributions is given by

$$f_{\alpha,f}(\mathbf{x}_{c\to f}) = f_\alpha^{eq}(\rho(\mathbf{x}_{c\to f}), \mathbf{u}(\mathbf{x}_{c\to f})) + \frac{\omega_c}{2\omega_f} f_{\alpha,c}^{neq}(\mathbf{x}_{c\to f}), \tag{2.43}$$

$$f_{\alpha,c}(\mathbf{x}_{f\to c}) = f_\alpha^{eq}(\rho(\mathbf{x}_{f\to c}), \mathbf{u}(\mathbf{x}_{f\to c})) + \frac{2\omega_f}{\omega_c} f_{\alpha,f}^{neq}(\mathbf{x}_{f\to c}). \tag{2.44}$$

Eq. 2.44 describes the coupling from the fine to the coarse grid. However, the small scales resolved by the fine grid can not be resolved by the coarse grid. Therefore, a filtering operation is applied to the non-equilibrium part of the fine grid distribution functions. The decision to filter only the non-equilibrium parts is made in order to avoid strong dissipation effects added by the filter when applied to either the complete distribution functions or the lattice density and velocity, as shown in Ricot *et al.* [106]. As pointed out by Lagrava *et al.* [54], filtering ρ and $\mathbf{u}$ results in artificial viscosity increase at the refinement region leading to unexpected behaviour of the system and decrease of the local accuracy. The filtering process proposed by Pellerin *et al.* [107] is applied to the non-equilibrium part of the distributions as

$$\begin{aligned}\bar{f}_i(\mathbf{x}, t) = 0.25 f_i(\mathbf{x}, t) &+ 0.125(f_i(\mathbf{x}+\mathbf{e}_1, t) + f_i(\mathbf{x}+\mathbf{e}_2, t) + f_i(\mathbf{x}+\mathbf{e}_3, t) \\ &+ f_i(\mathbf{x}+\mathbf{e}_4, t)) + 0.0625(f_i(\mathbf{x}+\mathbf{e}_5, t) + f_i(\mathbf{x}+\mathbf{e}_6, t) \\ &+ f_i(\mathbf{x}+\mathbf{e}_7, t) + f_i(\mathbf{x}+\mathbf{e}_8, t)),\end{aligned} \tag{2.45}$$

where the overbar denotes the filtered quantity. Proper filtering of the non-equilibrium distributions is crucial in order to suppress instabilities and spurious oscillation originating from transferring information to the coarse that can not be resolved. The filtering function in Eq. (2.45) has been selected because of its stability and simplicity and does not introduce any spurious effects or numerical dissipation.

Eq. 2.43 describes the coupling from the coarse to the fine grid. As shown in Fig. 2.6, there exist sites in the fine grid that do not have a corresponding coarse site, thus an interpolation process is needed in order to complete the missing distributions. A symmetric, cubic spline fitting is used for the spatial interpolation of ρ, $\mathbf{u}$ and f_{α}^{neq} in order to eliminate any spatial asymmetries.

$$f(x) = \alpha_i + \beta_i x + \gamma_i x^2 + \delta_i x^3, \quad x_{i-1} \leq x_i \leq x_{i+1}. \tag{2.46}$$

The tridiagonal systems of equations are solved using the Thomas algorithm under the restrictions of nodal continuity of the function $f(x)$ and its first and second derivatives, as well as zero second derivative $f^{''}(x)$ at the end nodes. The reader should refer to Tölke and Krafczyk [108] for a different approach. Recently, Geier *et al.* [109] introduced the use of second-order bubble functions for the velocity interpolation in order to achieve exact conservation of momentum at the interface. As the fine grid can resolve more scales than the coarse one, a deconvolution approach, as the one presented by Quéméré *et al.* [110] may be used in order to try and recreate the small scales when information is transferred from the coarse to the fine grid.

2.2.4 Grid refinement algorithm

The computational multi-domain algorithm used in this study is described below.

This algorithm describes the coupling between two grid resolution levels. If more resolution levels exist, a recursive algorithm based on the same principle of information exchange should be used. In this study, the computational domain is described with nine levels of refinement unless otherwise specified. Finally, using the nominal moment basis and the orthogonalised matrix $\mathbf{K}$ in Eq. (2.17),

Algorithm 1 Multi-Domain algorithm

```
1: Initialize $\rho$ and $\mathbf{u}$ on every domain and compute the equilibrium distributions $f_i^{eq}$.
2: Collide and Stream all distributions on the coarse grid. The coarse grid is now at time $t + \delta t_c$.
3: Collide and Stream once on the fine grid bringing it to time $t + \delta t_c/2$.
4: Perform a linear temporal interpolation of $\rho_c$, $\mathbf{u}_c$ and $f_{i,c}^{neq}$ at time $t + \delta t_c/2$ at the fine-coarse interface.
5: Spatially interpolate the values $\rho_c(t+\delta t_c/2)$, $\mathbf{u}_c(t+\delta t_c/2)$ and $f_{i,c}^{neq}(t+\delta t_c/2)$ at the fine nodes with no overlapping coarse nodes using Eq. (2.46).
6: All populations at the fine grid boundaries are reconstructed following a convective scaling using Eq. (2.43).
7: Collide and Stream once on the fine grid bringing it to time $t + \delta t_c$.
8: Spatially interpolate the values $\rho_c(t + \delta t_c)$, $\mathbf{u}_c(t + \delta t_c)$ and $f_{i,c}^{neq}(t + \delta t_c)$ as in step 5 using Eq. (2.46).
9: All populations at the fine grid boundaries are reconstructed following a convective scaling using Eq. (2.43).
10: Save the equilibrium and non-equilibrium distributions $f_{i,f}^{eq}$, $f_{i,f}^{neq}$ at the coarse-fine interface.
11: Replace all the populations at the coarse-fine interface using filtering and scaling, using Eq. (2.45) and Eq. (2.44).
12: Proceed to the next time-step. Go to step 2.
```

the equilibrium distribution may be computed as

$$f_0^{eq} = 4/9\rho - 2/3\rho(u_x^2 + u_y^2) + \rho u_x^2 u_y^2, \tag{2.47a}$$

$$f_1^{eq} = 1/9\rho + 1/3\rho u_x + 1/2\rho u_x^2 - 1/6\rho(u_x^2 + u_y^2) - 1/2\rho(u_x u_y^2 + u_x^2 u_y^2), \tag{2.47b}$$

$$f_2^{eq} = 1/9\rho + 1/3\rho u_y + 1/2\rho u_y^2 - 1/6\rho(u_x^2 + u_y^2) - 1/2\rho(u_y u_x^2 + u_x^2 u_y^2), \tag{2.47c}$$

$$f_3^{eq} = 1/9\rho - 1/3\rho u_x + 1/2\rho u_x^2 - 1/6\rho(u_x^2 + u_y^2) + 1/2\rho(u_x u_y^2 - u_x^2 u_y^2), \tag{2.47d}$$

$$f_4^{eq} = 1/9\rho - 1/3\rho u_y + 1/2\rho u_y^2 - 1/6\rho(u_x^2 + u_y^2) + 1/2\rho(u_y u_x^2 - u_x^2 u_y^2), \quad (2.47e)$$

$$f_5^{eq} = 1/36\rho + 1/12\rho(u_x + u_y + u_x^2 + u_y^2)$$
$$+ 1/4\rho(u_x u_y + u_x^2 u_y + u_x u_y^2 + u_x^2 u_y^2), \quad (2.47f)$$

$$f_6^{eq} = 1/36\rho + 1/12\rho(-u_x + u_y + u_x^2 + u_y^2)$$
$$+ 1/4\rho(-u_x u_y + u_x^2 u_y - u_x u_y^2 + u_x^2 u_y^2), \quad (2.47g)$$

$$f_7^{eq} = 1/36\rho + 1/12\rho(-u_x - u_y + u_x^2 + u_y^2)$$
$$+ 1/4\rho(u_x u_y - u_x^2 u_y - u_x u_y^2 + u_x^2 u_y^2), \quad (2.47h)$$

$$f_8^{eq} = 1/36\rho + 1/12\rho(u_x - u_y + u_x^2 + u_y^2)$$
$$+ 1/4\rho(-u_x u_y - u_x^2 u_y + u_x u_y^2 + u_x^2 u_y^2). \quad (2.47i)$$

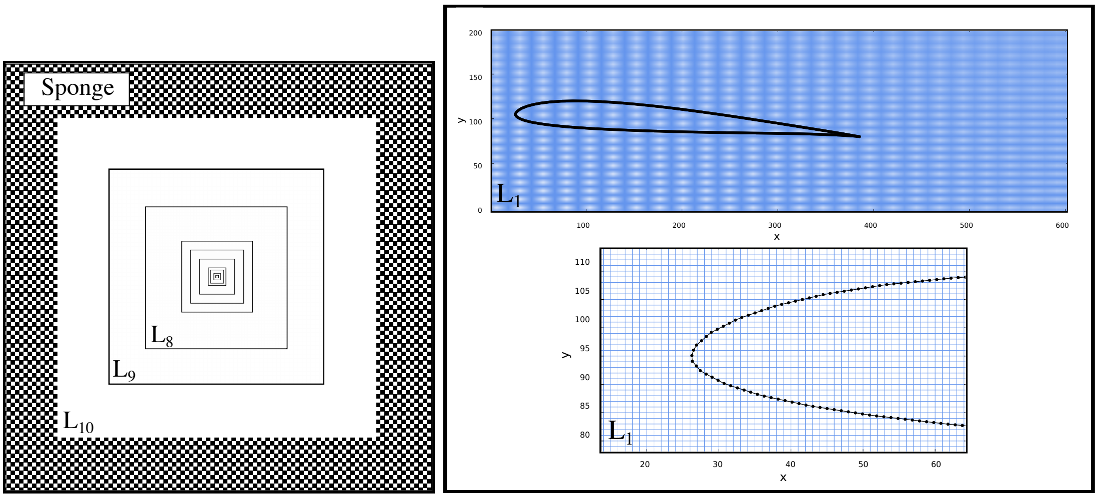
Note that the equilibrium distribution in the CLBM contains higher order velocity terms as compared to the standard LBM [27].

2.2.5 Grid layout

Figure 2.8 shows the arrangement of ten grid with different resolutions used for the simulation of the flow around an oscillating SD7003 airfoil with a chord length of $c = 360lu$. Please note that this is only an indicative example of the grid layouts used in this study. For different numerical examples the grid sizes are scaled accordingly to match the simulation set ups. This arrangement is not optimal in terms of computational efficiency. Table 2.1 shows the grid structures of different refinement levels. The relative grid sizes correspond to the finest grid resolution. Sponge layers are included in the L_{10} level.

2.3 Domain Boundary Conditions

Choosing appropriate boundary conditions (BCs) is of crucial significance in the LBM framework. The boundary conditions used in this study will be discussed



(a) (b)

Figure 2.8: (a) Computational grid with $L = 10$ levels of refinement, (b) Fine grid around the a SD7003 airfoil with $c = 360lu$.

Table 2.1: Grids levels, absolute and relative to the finest grid sizes.

Grid Level	Grid Size	Relative Grid Size
L_1	605×205	605×205
L_2	401×201	801×401
L_3	301×301	1201×1201
L_4	201×201	1601×1601
L_5	201×201	3201×3201
L_6	151×151	4831×4831
L_7	101×101	6463×6463
L_8	101×101	12927×12927
L_9	75×75	19201×19201
L_{10}	75×75	38401×38401

in this section. Unless specified otherwise, the boundary conditions described in the following sections have been used in all simulations presented in this thesis.

2.3.1 Description of the problem

Every LB algorithm consists of two major steps, the collision and the streaming of the populations. On the nodes next to the boundary some information come from the boundary nodes, and therefore will be non-physical. As shown in Fig. 2.9, the node located on the left boundary of the computational domain contains information coming for a node inside the wall (populations f_1, f_5 and f_8). In order to maintain the consistency of the numerical method, these population must be reconstructed. Populations f_0, f_2, f_3, f_4,f_6 and f_7 coming from the bulk are know. However, additional information are required in order to accurately reconstruct the remaining populations.

For simplicity, let us only consider flat wall BCs. Various ways of reconstructing the missing populations on the boundary nodes for either flat (see Zou and He [57], Inamuro [58], Latt and Chopard [59], Skordos [60], Ansumali and Karlin [61], Halliday *et al.* [62]) or curved boundaries (see Bouzidi [63], Ginzburg and d'Humières [38], Junk and Yang [26], Geier *et al.* [48]) can be found in the

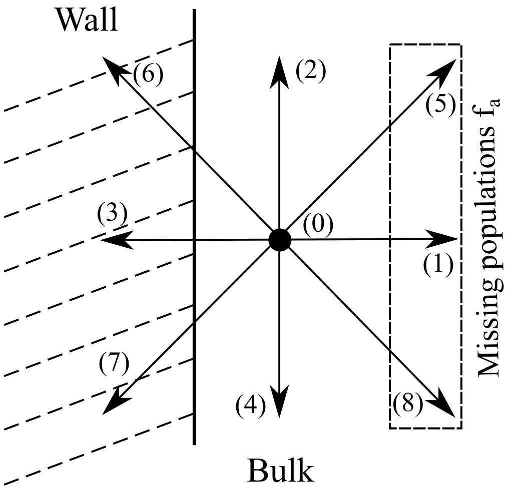


Figure 2.9: Missing (1,5,8) and known (0,2,3,4,6,7) populations on a boundary node after the streaming step.

literature. All the precited BCs share a common feature, they are all applied to the first bulk node close to the wall. Therefore, the reconstruction of the populations and the computation of the macroscopic variables ρ and $\mathbf{u}$ must be computed before the collision step that depends on the equilibrium distribution $f^{(eq)}(\rho, \mathbf{u})$. Let us assume that the velocity $\mathbf{u}$ is prescribed on the boundary (Dirichlet BC). In order to complete the reconstruction of the missing populations, the density on the node must be computed.

2.3.2 Density computation

Using Eq. 2.31, the density can be simply computed by summing the populations on the node. Following Zou and He [57], let us define a unit vector $\mathbf{n}$ normal to the wall pointing outwards of the computational domain [111]. In addition, the density ρ is decomposed into three parts $\rho_0 = \sum_{\alpha \in A_0} f_\alpha$, where $A_0 = \{\alpha | \mathbf{e}_\alpha \cdot \mathbf{n} = 0\}$, $\rho_+ = \sum_{\alpha \in A_+ f_\alpha}$, where $A_+ = \{\alpha | \mathbf{e}_\alpha \cdot \mathbf{n} > 0\}$ and $\rho_- = \sum_{\alpha \in A_- f_\alpha}$, where $A_- = \{\alpha | \mathbf{e}_\alpha \cdot \mathbf{n} < 0\}$. Using Eq. 2.31 and defining $u_\perp = \mathbf{u} \cdot \mathbf{n}$ as the projection of $\mathbf{u}$ on $\mathbf{n}$, we have $u_\perp = (\rho_+ - \rho_-)/\rho$. Therefore, the density ρ may be computed as

$$\rho = \frac{1}{1 + u_\perp}(2\rho_+ + \rho_0). \tag{2.48}$$

Therefore, for flat walls only, the density ρ can be computed since ρ_+ and ρ_0 are always known, as long as the condition $u_\perp = \mathbf{u} \cdot \mathbf{n}$ is imposed on the boundary. As a final remark, it should be noted that all the information required for the reconstruction of the missing populations is available as soon as both ρ and $\mathbf{u}$ are now known. In the precited BCs two approaches are used. In the first, only the missing populations are reconstructed, whereas in the second all populations are replaced using only the macroscopic information, such as the velocity, the density and higher-order moments [111]. The later is used in this study.

2.3.3 Inlet: Velocity boundary conditions

In this study, the method proposed by Latt and Chopard [59] is used for the inlet velocity boundary conditions, where all populations are reconstructed based on the knowledge of the macroscopic variables. The present scheme is used due to its superior numerical stability over other schemes at low viscosity values. Taking the second order raw moments of the distributions f_α let us define the tensor $\mathbf{\Pi} = \sum_{\alpha=0}^{q-1} \mathbf{e}_\alpha \mathbf{e}_\alpha f_\alpha$ and its non-equilibrium counterpart $\mathbf{\Pi}^{(1)} = \sum_{\alpha=0}^{q-1} \mathbf{e}_\alpha \mathbf{e}_\alpha f_\alpha^{(neq)}$. In addition, let us define the tensor $\mathbf{Q}_\alpha = \mathbf{e}_\alpha \mathbf{e}_\alpha - c_s^2 \mathbf{I}$, where $\mathbf{I}$ is the identity matrix.

First, the bounce back of off-equilibrium parts rule $f_\alpha^{(neq)} = f_{opp(\alpha)}^{(neq)}$ ([57]) is used in order to temporarily compute the unknown populations as $f_\alpha = f_\alpha^{(eq)}(\rho, \mathbf{u}) + f_{opp(\alpha)} - f_{opp(\alpha)}^{(eq)}$. Finally, following Latt and Chopard [59], all particle populations on the boundary nodes can be reconstructed as

$$\hat{f}_\alpha = f_\alpha^{(eq)}(\rho, \mathbf{u}) + \frac{w_\alpha}{2c_s^2} \mathbf{Q}_\alpha : \mathbf{\Pi}^{(1)}, \tag{2.49}$$

2.3.4 Outflow boundary conditions

The most common treatment in the literature for the outlet boundary condition is the use of a simple extrapolation scheme, where the missing populations are copied from the last node before the outlet. Assuming that the outlet is located in the positive x-direction of the domain $\mathbf{x}$, it can be seen from Fig. 2.9 that the populations f_3, f_6 and f_7 are missing. A simple approach to fill the missing populations would be to copy them from the neighbouring node at $\mathbf{x} - \delta x$ as

$$f_{(3,6,7)}(\mathbf{x}, t) = f_{(3,6,7)}(\mathbf{x} - \delta x, t). \tag{2.50}$$

However, it has been shown that this approach cause strong acoustic reflections that dissipate in the computational domain and affect the accuracy of the solution. In this study, an extrapolation scheme in space and time is applied for the missing distributions [48]. Assuming only waves normal to the boundary the missing distributions are computed as

$$f_\alpha(\mathbf{x},t) = f_\alpha(\mathbf{x}-\delta x, t-\delta t)\big(c_s-\mathbf{u}(\mathbf{x},t-\delta t\big)\delta t/\delta x+\Big(1-\big(c_s-\mathbf{u}(\mathbf{x},t-\delta t)\big)\Big)f_\alpha(\mathbf{x},t-\delta t). \tag{2.51}$$

2.3.5 Open boundaries: Free Slip

Open boundaries in the LBM framework are normally treated as periodic boundaries. Considering the case shown in Fig. 2.9, the missing populations $f_{(1,5,8)}(\mathbf{x}=0,t)$ at time t are filled with the known populations $f_{(1,5,8)}(\mathbf{x}=N_x,t-\delta t)$ at time $t-\delta t$. However, in order to avoid dissipation of spurious effects across the boundaries a free slip condition is used in this study, where the missing populations on a boundary node are filled with their mirror populations in the direction of the boundary. In the case of a boundary located at the left end of the domain, as shown in Fig. 2.9, the missing populations are filled as $f_1(\mathbf{x},t)=f_3(\mathbf{x},t)$, $f_5(\mathbf{x},t)=f_6(\mathbf{x},t)$ and $f_8(\mathbf{x},t)=f_7(\mathbf{x},t)$.

2.3.6 Non-reflecting boundary conditions

In LBM simulations, it is important that the boundaries interact with the bulk flow in a way that spurious signals coming from the propagation of acoustic waves or other sources are cancelled and the prescribed boundary conditions are satisfied. In order for the boundary to process the pressure change caused by the incoming wave and suppress the reflection of the wave back to the domain, a dynamic interaction between the boundary and the fluid is essential. However, satisfying both requirements is not always a trivial task. Schlaffer [112] presented a very coherent comparison of various non-reflecting boundary conditions in the LBM framework and showed that a consistent non-reflecting boundary condition leads to faster converge rates and suppresses the propagation of low-frequency spurious pressure waves.

Non-reflecting methods can be divided into three categories, filtering methods, characteristic based methods and impedance methods [113, 114, 115, 112]. Filtering methods employ either continuous or discrete low-pass filters and are applied directly on the discrete populations in order to “remove” the pressure wave. However, the efficiency of the method is not sufficient [112]. In contrast to filtering methods, characteristic based methods solve a simplified set of constitutive equations on the boundary by setting the amplitude of the acoustic waves to zero and assuming for an adjacent region outside the domain. A linear one dimensional inviscid (LODI) system of equations is solved by neglecting the derivatives normal to the boundary. The method is characterized by a distinct relation between the angle of incidence of the acoustic wave and its reflectance dependence. Overall, the method has been successfully applied to a few applications [113, 116, 112]. Finally, impedance boundary conditions use an integral formulation of the conservation quantities, vanishing the spatial derivatives emerging in the differential formulation (LODI method), maintaining a strict locality and increasing the numerical accuracy. Let us assume that every change in pressure propagates with the speed of sound. For a full derivation of impedance BCs for both compressible and incompressible LBM, the reader should refer to Schlaffer [112].

The significance of filtering the spurious effects originating from the interaction of the acoustic waves with the domain boundaries is high when a grid refinement scheme with acoustic scaling is used. Acoustic waves originating from the boundary located in the finest resolution grid propagate across grids with different resolution levels until reaching the boundaries of the domain. A result of using an acoustic scaling in a multi domain scheme is that as we move from finer to coarser resolutions the lattice kinematic viscosity decreases, as shown in section 2.4. This gradual decrease in the shear viscosity magnifies the spurious signals coming from the propagation of the acoustic waves. This effect is even stronger when high Reynolds number flows are investigated. It is important to note that the cascaded LBM can handle the very low viscosity values at the low resolution grids. However, we choose to filter all the artificial spurious acoustic effects at the boundaries in order to avoid any undesirable effects that could compromise the accuracy of the solution. It should be noted that since the domain boundaries are very coarse in the present multi-domain scheme, the implementation of non-reflecting boundary conditions does not add any considerable computational overhead.

In order to adapt the impedance for an arbitrary direction of the incoming sound pressure wave, an isotropic formulation is considered in this study. First, let us define the rate of momentum change per unit area as $du^{(n)} = u^{(n)} - u^{(n-1)}$, where n is the current time step. In order to project $du^{(n)}$ onto the corresponding direction of incidence, it is expressed by its absolute value and corresponding sign as

$$\|du\| = \sqrt{du_x^2 + du_y^2} = \sqrt{\left(u_x^{(n)} - u_x^{(n-1)}\right)^2 + \left(u_y^{(n)} - u_y^{(n-1)}\right)^2} \tag{2.52}$$

and $du = \text{sign}(du)\|du\|$. For the left boundary shown in Fig. 2.9, and using the symmetry conditions of the D2Q9 lattice the following condition is then imposed

$$\begin{aligned} I_c := \mp\left(\rho^{(n)} - \rho^{(n-1)}\right)c_s^2 \mp \frac{1}{2}\rho^{(n)}\left[\left(u_x^{(n)} - u_x^{(n-1)}\right)^2 + \left(u_y^{(n)} - u_y^{(n-1)}\right)^2\right] \\ + \text{sign}(du_x)\sqrt{\left(u_x^{(n)} - u_x^{(n-1)}\right)^2 + \left(u_y^{(n)} - u_y^{(n-1)}\right)^2}\rho^{(n)}c_s = 0 \end{aligned} \tag{2.53}$$

Using the macroscopic variables in Eq. 2.31 without the external forces as well as Eq. 2.48, the current density $\rho^{(n)}$ on the boundary is expressed as $\rho^{(n)} = \rho_z/(1 - u_x^{(n)})$, where $\rho_z = f_0 + f_2 + f_4 + 2(f_3 + f_6 + f_7)$. The vertical velocity $u_y^{(n)}$ is computed as $u_y^{(n)} = 3(f_2 - f_4)/2\rho^{(n)}$using the bounce back of the non-equilibrium parts rule. The Newton-Raphson iterative algorithm is used for the solution of Eq. 2.53 and the computation of $u_x^{(n)}$. The sign of du_x can be computed using the following equation

$$\begin{aligned} u_x^{(n)} = u_x^{(n-1)} &- \left(c_s^2 \frac{\rho^{(n-1)}}{\rho_z} + c_s \right) \\ &+ \sqrt{\left(c_s^2 \frac{\rho^{(n-1)}}{\rho_z} + c_s \right)^2 - 2c_s^2 \left(\frac{\rho^{(n-1)}}{\rho_z} \left(u_x^{(n-1)} - 1 \right) + 1 \right)}. \end{aligned} \tag{2.54}$$

Once the macroscopic density and velocity are computed, the regularized scheme in Section 2.3.3 and the extrapolation scheme in Section 2.3.4 are used to compute the missing populations on the left and right boundaries respectively.

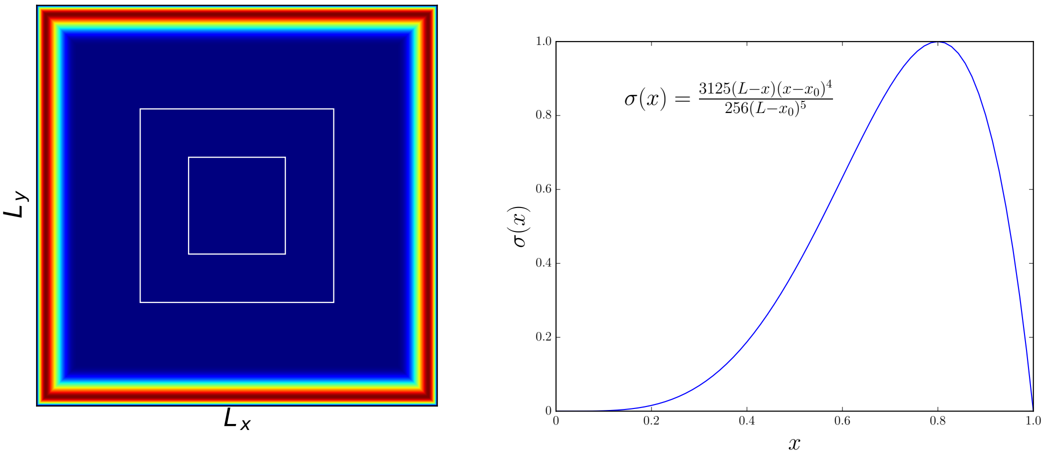
2.3.7 Absorbing layers for the LBM

More recently, Xu and Sagaut [117] addressed the issue of absorbing acoustic boundary conditions for the LBM, in order to damp and minimise the acoustic wave reflections from the computational boundaries. Effectively, an absorbing term is added to the right hand side (RHS) of the governing fluid equations, [117, 118]. In this study, the Type II absorbing term proposed by Xu and Sagaut [117] is implemented in the cascaded LBM.

First, the right hand side of Eq.2.25 need to be modified by adding the absorbing term $H_a^{(eq)}$ as

$$f_\alpha(\mathbf{x} + \mathbf{e}_\alpha \delta t, t + \delta t) = f_\alpha(\mathbf{x}, t) + \Omega_a^*(\mathbf{x}, t) + \delta t S_a + \delta t H_\alpha^{eq}(\rho^f, \mathbf{u}^f, \rho^*, \mathbf{u}^*, t), \tag{2.55}$$

where $H_\alpha^{eq}(\rho^f, \mathbf{u}^f, \rho^*, \mathbf{u}^*, t) = \chi\Big(f_\alpha^{eq}(\rho^f, \mathbf{u}^f, t) - f_\alpha^{eq}(\rho^*, \mathbf{u}^*, t)\Big)$ with $\chi = \sigma(x)$ being the strength of the absorbing layer, as shown in Fig. 2.10b.



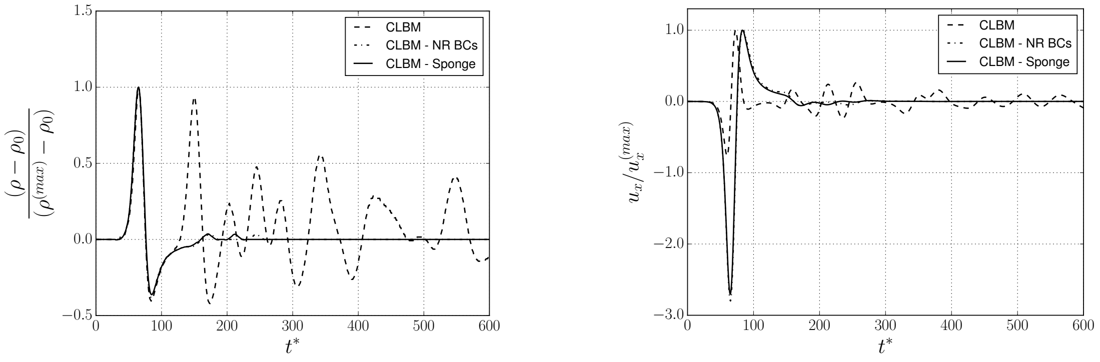
(a) **(b)**

Figure 2.10: (a): 2D computational domain with absorbing layer. (b): Normalized absorbing strength profile $\sigma(x)$ with $x_0 = 0$ and $L = 1$.

The superscript f denotes the farfield values of the velocity and density, whereas the superscript $*$ denotes the parametrized density $\rho^* = \sum_\alpha f_\alpha + n\delta t \sum_\alpha H_\alpha^{eq}(\rho^f, \mathbf{u}^f, \rho, \mathbf{u}, t)$ and velocity $\rho^* u_j^* = \sum_\alpha e_{\alpha j} f_\alpha + m\delta t \sum_\alpha e_{\alpha j} H_\alpha^{eq}(\rho^f, \mathbf{u}^f, \rho, \mathbf{u}, t)$ [96]. $m = n = 1/2$ as in [117]. In a 2D Domain, Fig. 2.10a the absorbing layer is effectively a "sponge" that damps the acoustic wave reflections.

2.3.8 Test case: Acoustic pressure wave propagation

The effectiveness of the non-reflecting BCs and the absorbing layer treatment on removing spurious signals coming from the propagation of acoustic waves is investigated in this section. A 2D acoustic Gaussian pulse source is considered originating for the center (x_0, y_0) of a 400×400 computational domain. Periodic boundary conditions have been used. The kinematic viscosity is set at $\nu = 0.5$. The initial flow profile is given as $\rho_0 = 1 + \rho'$ and $(u_x, u_y) = (0, 0)$. The density



(a) **(b)**

Figure 2.11: Reflection of acoustic wave at the left boundary. (a) Reflection of density and (b) reflection of normal to the wall velocity.

fluctuation $\rho^{'}$ is given by

$$\rho^{'} = \epsilon exp(-\alpha \cdot r^2), \tag{2.56}$$

where $\epsilon = 10^{-3}$, $\alpha = ln(2)/b^2$ with $b = 1/20$ and $r = \sqrt{(x-x_0)^2 + (y-y_0)^2}$. Periodic boundary conditions are applied to all boundaries.

Three cases are investigated: (i) no treatment is applied on the boundaries, (ii) the non-reflecting boundary conditions of Schlaffer [112] described in 2.5.6 are applied to all periodic boundaries and (iii) the absorbing layer scheme of Xu and Sagaut [117], as shown in Fig. 2.10a, encloses the computational domain. Fig. 2.11 shows the reflection of the acoustic waves at the left boundary for all three cases. As observed, when the waves are allowed to reflect from the boundaries (Case 1), strong fluctuations occur in both the density and the velocity. However, when absorbing conditions are used (Cases 2 and 3) a steady state is achieved immediately after the first interaction of the wave with the boundary and all fluctuations are significantly suppressed. Similar results are shown in Fig. 2.12, where a steady density state $\rho \approx \rho_0$ is obtained in the computational domain in Case 2 and Case 3, whereas in Case 1 the acoustic waves keep reflecting from the

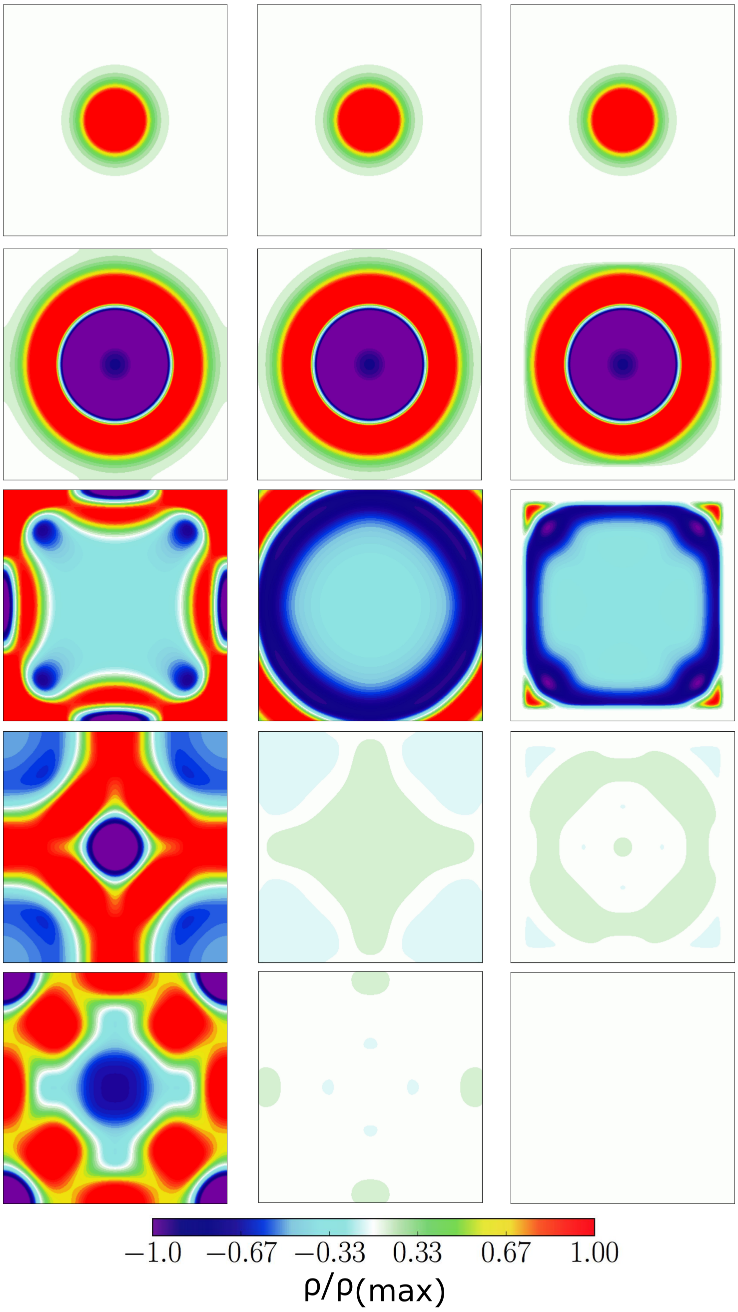


Figure 2.12: Pressure propagation (a) without treatment - left (b) with non-reflecting BCs - middle and (c) with the sponge absorbing layer -right.

boundaries, slowly wearing off. In the present study, the treatment of Xu and Sagaut [117] is used.

Chapter 3

The Immersed Boundary Method

In this section, the mathematical basis of the immersed boundary method will be discussed. That includes the continuous and discrete formulation of the IBM, as well as the kernel functions and their significance in the accuracy of the method. A detailed review of the most popular schemes for evaluating the force density on the boundary is included. The performance properties of the presented schemes is briefly discussed. An extensive comparison of the accuracy of two iterative IB schemes coupled with the CLBM will be presented in Chapter 6. Finally, a modified Lagrange point approximation scheme in order to accurately capture the effects of the internal mass is derived.

As shown in Chapter 2, boundary conditions in the LBM framework usually modify and/or reconstruct the populations f_α, in order to obtain the desired values for the velocity and pressure at the boundary. In the immersed boundary method (IBM) framework, the boundary condition is prescribed by adding a force density $\mathbf{F}(\mathbf{x}, t)$ to the governing fluid equations. To this end, the computation of $\mathbf{F}(\mathbf{x}, t)$ describes a fluid that behaves as a boundary with desired properties [119]. In Peskin's original derivation of the IBM [65], the force density $\mathbf{F}(\mathbf{x}, t)$ is added to the Navier-Stokes equation. A boundary condition prescribed by the IBM exists on the macroscopic level and therefore is not aware of the populations f_α. Mathematically, the basis of the immersed boundary method (IBM) includes first, an Eulerian grid and a set of Lagrangian markers and second, a velocity interpolation and a force spreading operation as devised originally by Peskin [65]. The IBM was first combined with the LBM by Feng and Michaelides [99] and has since then become a popular approach.

3.1 Introduction to the continuous immersed boundary equations

Consider a closed curve $\mathscr{E}(t)$ immersed in a two-dimensional fluid domain $\Omega(t)$ as shown in Fig. 3.1a. Essentially, the geometry is decomposed into two coordinates systems. The closed curve $\mathscr{E}(t)$ is represented by an ensemble of marker points (Lagrangian) $\mathbf{X}_{\mathscr{L}}(s,t)$ that are allowed to move freely in the fluid and are independent of the Eulerian grid, defined by the $\mathbf{x}$ coordinates. Theoretically, the relative distances between the Lagrangian markers should be kept constant, $|\mathbf{X}_{\mathscr{L}}(s,t)-\mathbf{X}_{\mathscr{L}}(l,t)| \approx$ *const*. In fully coupled fluid structure interaction problems, a *nonconforming* surface mesh that defines the markers and their connectivity is required. Finally, a bi-directional coupling between the fluid and the boundary must be prescribed.

For viscous incompressible flows, the governing equations of the immersed boundary formulation are formulated as

$$\nabla \cdot \mathbf{u} = 0, \tag{3.1}$$

$$\rho(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}) + \nabla p = \mu \triangle \mathbf{u} + \mathbf{g}, \tag{3.2}$$

$$\mathbf{g}(\mathbf{x},t) = \int\limits_{\mathscr{E}} \mathbf{G}(s,t)\delta(\mathbf{x} - \mathbf{X}_{\mathscr{L}}(s,t))ds, \tag{3.3}$$

$$\frac{\partial \mathbf{X}_{\mathscr{L}}(s,t)}{\partial t} = \mathbf{U}(\mathbf{X}_{\mathscr{L}}(s,t),t) = \int\limits_{\mathscr{E}} \mathbf{u}(\mathbf{x},t)\delta(\mathbf{x} - \mathbf{X}_{\mathscr{L}}(s,t))d\mathbf{x}, \tag{3.4}$$

$$\mathbf{G}(s,t) = \mathbf{S}(\mathbf{X}_{\mathscr{L}}(s,t),t). \tag{3.5}$$

Eqs. (3.1 - 3.2) are the Eulerian Navier-Stokes equations with external forces $\mathbf{g}$ for the fluid domain $\Omega(t)$, whereas Eqs. (3.3 - 3.5) are the equations in Lagrangian formalism for the immersed boundary $\mathscr{E}(t)$. $\mathbf{X}_{\mathscr{L}}$, $\mathbf{U}$, $\mathbf{G}$ are the position of the IB in Lagrangian coordinates, the IB velocity and force density, respectively. $\mathbf{x}$, $\mathbf{u}$, $\mathbf{g}$, ρ and p are the Cartesian coordinates, fluid velocity, external force density, density and pressure, respectively. $\delta(\mathbf{x} - \mathbf{X}_{\mathscr{L}})$ is the Dirac's delta distribution function defined as

$$\delta_h(\mathbf{x} - \mathbf{X}_{\mathscr{L}}) = \frac{1}{h^3}\phi\left(\frac{x - X_{\mathscr{L}}}{h}\right)\left(\frac{y - Y_{\mathscr{L}}}{h}\right)\left(\frac{z - Z_{\mathscr{L}}}{h}\right), \tag{3.6}$$

where $\phi(\frac{x-X}{h})/h$ is the one-dimensional discrete delta function. $X_{\mathscr{L}}$, $Y_{\mathscr{L}}$ and $Z_{\mathscr{L}}$ are the Lagrangian components of $\mathbf{X}_{\mathscr{L}}$ whereas, x, y and z are the Eulerian components of $\mathbf{x}$.

3.2 The discrete immersed boundary equations

Let us now focus on the discretisation of the immersed boundary equations, Eqs. (3.3 - 3.4). The surface integrals in Eqs. (3.3 - 3.4) cannot be exactly computed since both the velocity and force fields are only known at the discrete lattice cites. Therefore, the integrals are replaced by summations over the ensemble of Lagrangian points. The delta distributions are also discretised. First, let us assume that the fluid in the entire domain Ω is homogeneous and second, that the Lagrangian markers are massless. The latter assumption indicates that the immersed boundary has the same density as the surrounding fluid. The discrete

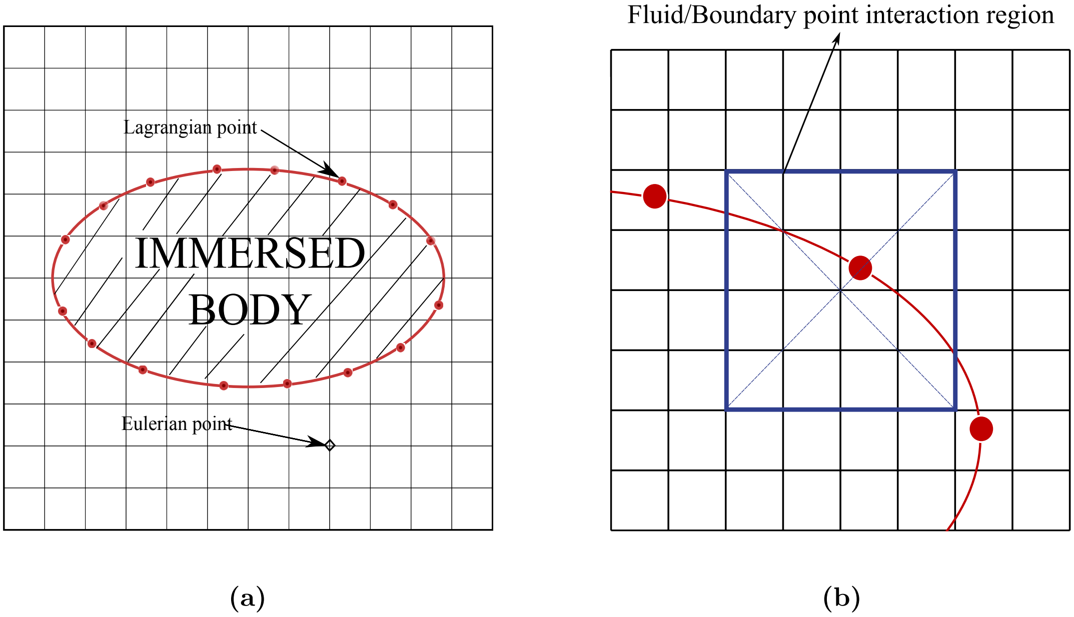


Figure 3.1: (a): Immersed boundary illustration using a set of Lagrangian points to represent the boundary and a set of Eulerian points defined by the intersection points of the mesh lines to represent the fluid domain. (b): Fluid/Boundary interaction regions where the kernel functions are applied.

IBM equations are

$$\mathbf{u}_{l,t} = \sum_{i,j} \mathbf{u}_{ij,t} D(\mathbf{x}_{ij,t} - \mathbf{x}_{l,t}) h^2, \tag{3.7}$$

$$\mathbf{F}_{ij,t} = \sum_{l} \mathbf{F}_{l,t} D(\mathbf{x}_{ij,t} - \mathbf{x}_{l,t}) \Delta s_l. \tag{3.8}$$

For the full derivation of the general IBM set of equations see Peskin [65]. In the discrete formulation the fluid is discretised as an Eulerian lattice with coordinates $\mathbf{x}_{ij}$ and the boundary as a set of Lagrangian markers at $\mathbf{x}_l$. Eq. (3.7) computes the interpolated boundary velocity $\mathbf{u}_l$ from the surrounding Eulerian velocity $\mathbf{u}_{ij}$. In Eq. (3.8) the total computed force acting on the markers $\mathbf{F}_l$ is spread to the surrounding Eulerian grid ($\mathbf{F}_{ij}$ is the force density on the lattice). It is important to note that the total force $\mathbf{F}_l$ is first computed in the Lagrangian markers and then spread to the lattice. Δs_l is the discretisation length of the boundary which must satisfy $\Delta s_l = \Delta s_0$. At this point, the key questions that rise are, (i) what is the most appropriate form of the kernel function $D(\mathbf{x}_{ij,t} - \mathbf{x}_{l,t})$, (ii) how the $\mathbf{F}_l$ can be computed, and (iii) how is the force density $\mathbf{F}_{ij}$ incorporated in the LBM to update the fluid.

3.3 The kernel functions

3.3.1 Fundamental principles

First, it should be noted that the consturction of the discrete delta function in Eq. (3.6) is not unique (see Peskin [65] for a detailed description of appropriate discretisation procedures). Using Eq. (3.8), and considering the integral conservation of the force spread onto the grid and its first moment, the following restrictions must hold

$$\sum_l \mathbf{F}(\mathbf{x}_l, t)\Delta s_l = \sum_{i,j} \mathbf{F}(\mathbf{x}_{i,j}, t)h_{i,j}, \tag{3.9a}$$

$$\sum_l \mathbf{x}_l \times \mathbf{F}(\mathbf{x}_l, t)\Delta s_l = \sum_{i,j} \mathbf{x}_{i,j} \times \mathbf{F}(\mathbf{x}_{i,j}, t)h_{i,j}. \tag{3.9b}$$

These properties ensure that the forces and moments exerted on the solid by the fluid may be accurately computed by integrating the force field [120]. For the uniform Cartesian grid $h = h_i = h_j$ used in this study, the discrete delta function must have the following properties in order for Eqns. (3.9a - 3.9b) to be verified

$$\sum_{i,j} D(\mathbf{x}_{ij,t} - \mathbf{x}_{l,t})h_{ij}^2 = 1, \tag{3.10a}$$

$$\sum_{i,j} (\mathbf{x}_{ij,t} - \mathbf{x}_{l,t})D(\mathbf{x}_{ij,t} - \mathbf{x}_{l,t})h_{ij}^2 = 0. \tag{3.10b}$$

Note that, using Eq. (3.6), the discrete 2-dimensional kernel function $D(\mathbf{x}_{ij,t} - \mathbf{x}_{l,t})$ can be written as a product of 1-dimensional functions $D(\mathbf{x}_{ij,t} - \mathbf{x}_{l,t}) = D(x_{i,t} -$

$\mathbf{x}_{l,t})D(x_{j,t}-\mathbf{x}_{l,t})/h^2$. Eq. (3.10a) effectively imposes a normalisation of the kernel function. The above restrictions guarantee that both the momentum and the angular momentum will be identical when computed in either the Eulerian or the Lagrangian reference frame. In addition, in order to reduce the computational overhead of Eqns. ((3.7) - (3.8)) while maintaining the desired accuracy, the number of summands must be optimised. Finally, the delta function should suppress lattice artefacts such as interpolation "bumpiness" in moving boundary applications.

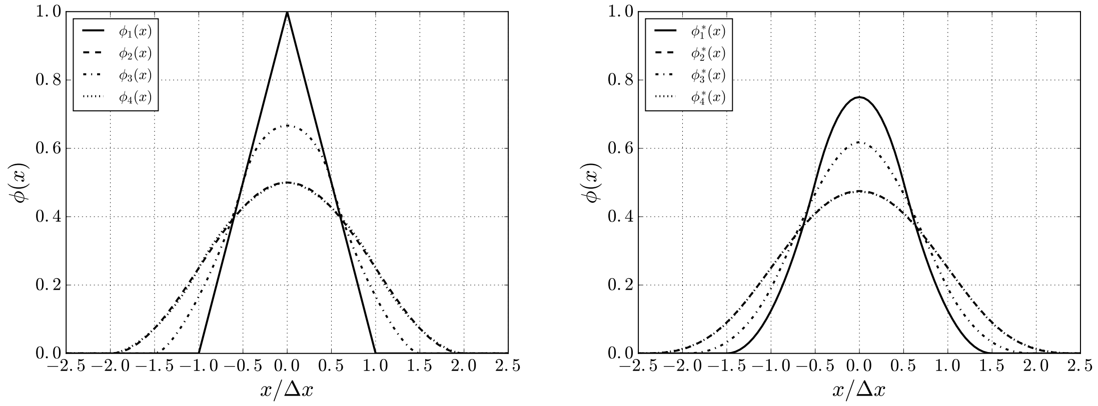
3.3.2 Smoothing technique

In order to suppress the non-physical oscillations in the computed forces, the method proposed by Yang *et al.* [121] is used in this study. It has been shown that the derivatives of regular discrete delta functions fail to satisfy certain moment conditions, leading to non-physical oscillations that affect both the solution accuracy and the convergence rate of the computation. The proposed smoothed functions have one-order higher derivative than the regular ones. In addition, the first two discrete moment conditions are satisfied and finally, their derivatives satisfy one-order higher moment condition than the regular ones.

Following the smoothing technique proposed by Yang *et al.* [121], a new function $\phi^*(x)$ can be defined for the one-dimensional function $\phi(x)$ as

$$\phi^*\left(\frac{x_j - X_{\mathscr{L}}}{h}\right) = \frac{1}{h}\int\limits_{x_j-0.5h}^{x_j+0.5h} \phi\left(\frac{x' - X'}{h}\right) dx' \tag{3.11}$$

The original function has one lower derivative than the smoothed one. Finally introducing the transformations $r = (x_j - X)h^{-1}$ and $r' = (x' - X)h^{-1}$ the



(a) **(b)**

Figure 3.2: (a): Interpolation Kernels ϕ_1, ϕ_2, ϕ_3, ϕ_4. The kernel range is two, four, three and four lattice sites, respectively. (b): Smoothed Interpolation Kernels ϕ_1^*, ϕ_2^*, ϕ_3^*, ϕ_4^*. The kernel range is three, five, four and five lattice sites, respectively. It is observed that ϕ_2 and ϕ_2^* are very similar in shape to the respective ϕ_4 and ϕ_4^* kernels.

smoothed function can be rewritten in the following form

$$\phi^*(r) = \int_{r-0.5}^{r+0.5} \phi(r)dr' \tag{3.12}$$

Fig. 3.2 shows the most commonly used discrete delta functions in the IBM, as well as their respective smooth formulations. It can be seen that the smooth functions have wider supports than the regular ones. For a detailed analysis on the relevant properties of the smoothed functions the reader should refer to Yang *et al.* [121] . In this study, the 3-point and the 4-point piecewise discrete functions have been tested.

$$\phi_3(r) = \begin{cases} \frac{1}{3}(1 + \sqrt{-3r^2 + 1} & \text{if } |r| \leq 0.5 \\ \frac{1}{6}(5 - 3|r| - \sqrt{-3(1 - |r|)^2} & \text{if } 0.5 \leq |r| \leq 1.5 \\ 0 & \text{if } 1.5 \leq |r| \end{cases}, \tag{3.13a}$$

$$\phi_4(r) = \begin{cases} \frac{1}{8}(3 - 2|r| + \sqrt{1 + 4|r| - 4r^2} & \text{if } |r| \leq 1 \\ \frac{1}{8}(5 - 2|r| - \sqrt{-7 + 12|r| - 4r^2} & \text{if } 1 \leq |r| \leq 2 \\ 0 & \text{if } 2 \leq |r| \end{cases} . \tag{3.13b}$$

Using Eq. (3.12) their respective smoothed formulations are

$$\phi_3^*(r) = \begin{cases} \frac{17}{48} + \frac{\sqrt{3}\pi}{108} + \frac{|r|}{4} - \frac{r^2}{4} + \frac{1-2|r|}{16}\sqrt{+1 + 12|r| - 12r^2} & \\ -\frac{\sqrt{3}}{12}\arcsin(\frac{\sqrt{3}}{2}(2|r| - 1)) & \text{if } |r| \leq 1.0 \\ \frac{55}{48} - \frac{\sqrt{3}\pi}{108} - \frac{13|r|}{12} + \frac{r^2}{4} + \frac{2|r|-3}{48}\sqrt{-23 + 36|r| - 12r^2} & \\ -\frac{\sqrt{3}}{36}\arcsin(\frac{\sqrt{3}}{2}(2|r| - 3)) & \text{if } 1.0 \leq |r| \leq 2.0 \\ 0 & \text{if } 2.0 \leq |r| \end{cases} , \tag{3.14a}$$

$$\phi_4^*(r) = \begin{cases} \frac{3}{8} + \frac{\pi}{32} - \frac{r^2}{4} & \text{if } |r| \leq 0.5 \\ \frac{1}{4} + \frac{1-|r|}{8}\sqrt{-2 + 8|r| - 4r^2} - \frac{1}{8}\arcsin(\sqrt{2}(|r| - 1)) & \text{if } 0.5 \leq |r| \leq 1.5 \\ \frac{17}{16} - \frac{\pi}{64} - \frac{3|r|}{4} + \frac{r^2}{8} + \frac{|r|-2}{16}\sqrt{-14 + 16|r| - 4r^2} & \\ -\frac{1}{16}\arcsin(\sqrt{2}(|r| - 2)) & \text{if } 1.5 \leq |r| \leq 2.5 \\ 0 & \text{if } 2.5 \leq |r| \end{cases} . \tag{3.14b}$$

It has been verified that the use of the smoothed formulation significantly reduced the round-off errors caused by the interpolation procedures, especially in moving boundary flows where, the cumulative effect of those errors is critical to the overall accuracy.

3.3.3 Significance of interpolation accuracy in the IBM

Cheng *et al.* [122] investigated the effect of the discrete delta functions on the accuracy of the IBM coupled with the SRT-LBM. First, the relaxation of a circular membrane was examined. It has been found that the scheme with the $\phi_4(r)$ function leads to better volume conservation and pressure maintenance than others. In addition, the results revealed a surprising accuracy of the coupling scheme for transversal flow across the boundary [122]. Other functions with wider base stencils resulted in higher numerical errors. Therefore, it may be concluded that using functions with wider base stencils does not necessarily guarantee improvements in the interpolation accuracy.

A significant observation in the study of Cheng *et al.* [122] is the existence of numerical velocity slip near the immersed boundary. A similar conclusion is reported in [123]. The authors concluded that the errors are sensitive not only to the viscosity of the fluid but also to the discrete delta function used. In fact, for lower viscosity values, functions with wider base stencils resulted in more accurate results. Overall, the $\phi_4(r)$ function has superior accuracy and a moderate near-boundary influence domain. In order to suppress the velocity slip near the boundaries, the smoothed 4-point piecewise discrete function is used only for the force spreading operation in the IBM. In the velocity interpolation process the Lagrange polynomial is used as

$$U_l^{\mathscr{L}}(\mathbf{X}_l, t) = \sum_{ij} \left(\prod_{m=1, m\neq i}^{i_{max}} \frac{X_l - x_{mj}}{x_{ij} - x_{mj}} \right) \left(\prod_{n=1, n\neq j}^{j_{max}} \frac{Y_l - y_{in}}{y_{ij} - y_{in}} \right) u_{ij}(\mathbf{x} + \mathbf{e}_a \delta t, t + \delta t). \tag{3.15}$$

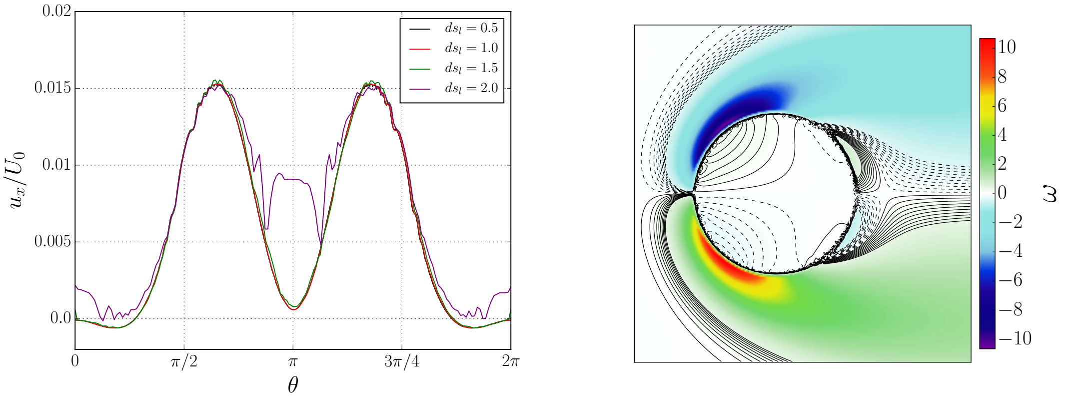
The use of the Lagrange polynomial function in the velocity interpolation stage seems to significantly decrease the numerical velocity slip (the computed velocity on the boundary does not exceed the analytical values). In IB-LBM simulations where non-slip boundary conditions are prescribed on the immersed bodies, fictitious flow patterns appear inside the solid domain. Numerical velocity slip is observed when the fictitious flow affects the numerical accuracy of the non-slip boundary condition. Contrary to across boundary errors caused by leakage of the flow across the boundary, velocity slip results in tangential boundary errors. Le and Zhang [123] related the velocity slip introduced by the IB-LBGK scheme to the lattice kinematic viscosity and derived a formula this artificial effect. As shown by Zhang *et al.* [103], at strong shearing high viscous flows, the interpolated velocity near the boundary using Eq. (3.16) is almost identical to the theoretical values.

Theoretically, this approach violates the condition where both the interpolation and the spreading steps must be performed using the same kernel. However, if the kernel range in both steps is similar, i.e. the force from a certain Lagrangian node is spread on the same grid nodes that were used for the interpolation of the velocity on that node, the present approach could be considered acceptable. In the investigated cases the use of different kernels in the IBM does not introduce any artificial effects that would decrease the solution accuracy. Further investigation is required in order to quantify the effects of using different kernels in immersed boundary methods.

3.3.4 Sensitivity to boundary discretization parameters

In order to prevent fluid leakage across the boundary, the discretisation length Δs_l must be carefully considered. As pointed out be Peskin [65], $\Delta s_l \leq h/2$ is necessary in order to guarantee the non-slip boundary condition on the boundary. Cheng *et al.* [122] found that if $\Delta s_l \leq 1h$ the errors are small and that further decreasing Δs_l does not significantly improve the accuracy of the solution. Between $1h \leq \Delta s_l \leq 2h$ the errors increase slightly as Δs_l increases, whereas the errors increase significantly for $\Delta s_l > 2h$. In our preliminary tests, it has been observed that the effect of the discretisation length Δs_l on the numerical accuracy of the IBM strongly depends on the flow characteristics, e.g. kinematic viscosity, as well as on the “sharpness” of the velocity profile on the boundary (boundary layer). In this study the discretisation length is set as $\Delta s_l \leq 1$, unless specified otherwise.

Fig. 3.3 illustrates the significance of the boundary discretization parameters on the computed flow around a stationary circular cylinder at $Re = 40$. The domain



(a) (b)

Figure 3.3: (a): Instantaneous interpolated surface velocity on a stationary circular cylinder at $Re = 40$ for $\Delta s_l = 0.5, 1.0, 1.5, 2.0$, (b) Instantaneous vorticity around the cylinder for $\Delta s_l = 2.0$.

configuration and the simulation parameters for this case are described in Chapter 6. For $\Delta s_l = 2.0h$, the interpolated boundary velocity is distorted, Fig. 3.3a. As shown in Fig. 3.3b, there is flow penetration throughout the immersed cylinder. That leads to significantly underestimated drag coefficient values compared to the values computed with $\Delta s_l = 0.5, 1.0, 1.5$.

3.4 Force density evaluation

In this section, the most commonly used schemes for computing the nodal forces on rigid boundaries will be discussed. Only the required steps for obtaining the force density on the boundary nodes will be discussed. The coupling between the IBM and the cascaded LBM, including the construction of accurate discrete forcing terms, will be presented in Chapter 4. In this study only two of the following schemes will be used: (i) the iterative direct forcing (IDF) method and (ii) the multi-direct forcing (MDF) scheme. However, for the sake of completeness some alternative algorithms will be presented as well.

It should be noted that in this study, only flows around stationary rigid bodies or rigid bodies undergoing prescribed motions will be examined. For deformable bodies simulations, two additional elements must be considered: (i) a constitutive model containing all the physics of the boundary deformation and (b) a surface mesh to evaluate the boundary deformation [119]. The interested reader should refer to Kruger *et al.* [124] and Sui *et al.* [125] for detailed model definitions. Sui *et al.* [125] present a coherent method of addressing both hyperelastic and viscoelastic models.

3.4.1 Explicit feedback IBM - The Penalty method

The penalty method has been proposed by Feng and Michaelides [99] in order to model the interaction between the fluid and the solid boundary. Let us assume that each boundary point $\mathbf{X}_l$ is individually connected by an elastic spring to its specific reference location $\mathbf{X}_l^*$. In this approach, the boundary nodes are slightly deformed from the interaction with the fluid as $\xi_l = \mathbf{X}_l - \mathbf{X}_l^*$. The displacement χ_l generates a force F_l that tends to restore the boundary back to its original position. The restoration force is modelled by a linear spring relation as

$$\mathbf{F}_l(t) = -\kappa \frac{\xi_l(t)}{|\xi_l(t)|}, \tag{3.16}$$

where κ is an empirically defined spring constant. Considering a steady flow, after several time steps the reference boundary points $\mathbf{X}_l^*$ will reach an “equilibrium” state and the computed force will be just enough to keep the boundary in place, locally satisfying the no-slip BC. The method is very sensitive as the optimum range of the parameter κ must be obtained and often requires a significantly small time-step. It has been shown that it is not possible to sufficiently satisfy the no-slip BC with an explicit IBM scheme.

3.4.2 Direct forcing methods

In order to overcome the main disadvantage of the explicit penalty method and improve the numerical accuracy and stability, direct forcing methods have been developed, where the boundary forces are directly computed from the flow field.

First let us express the desired velocities on the boundary nodes as $\dot{\mathbf{X}}_l = \mathbf{U}(\mathbf{X}_l)$. The fluid velocity $\mathbf{u}(\mathbf{x}, t)$ is computed after the streaming step in the LBM using Eq. (2.32) without the presence of external forces as

$$\mathbf{u}(\mathbf{x}, t) = \frac{1}{\rho(\mathbf{x}, t)} \sum_{\alpha}^{q} \bar{f}_\alpha(\mathbf{x}, t)\mathbf{e}_\alpha. \tag{3.17}$$

During the collision step, the presence of the body forces $\mathbf{F}(\mathbf{x}, t)$ updates the fluid velocity as

$$\mathbf{u}^p(\mathbf{x}, t) = \frac{1}{\rho(\mathbf{x}, t)} \sum_{\alpha}^{q} \bar{f}^p_\alpha(\mathbf{x}, t)\mathbf{e}_\alpha = \frac{1}{\rho(\mathbf{x}, t)} \sum_{\alpha}^{q} \bar{f}_\alpha(\mathbf{x}, t)\mathbf{e}_\alpha + \frac{\mathbf{F}(\mathbf{x}, t)\Delta t}{\rho(\mathbf{x}, t)}, \tag{3.18}$$

where the superscript p denotes the post-collision values. According to Guo *et al.* [96] the physical fluid velocity at time t may be computed using Eqns. (3.17) - (3.18) as an average of the pre and post collision values [119] as

$$\mathbf{u}^*(\mathbf{x}, t) = \frac{1}{\rho(\mathbf{x}, t)} \sum_{\alpha}^{q} \bar{f}_\alpha(\mathbf{x}, t)\mathbf{e}_\alpha + \frac{\mathbf{F}(\mathbf{x}, t)\Delta t}{2\rho(\mathbf{x}, t)}. \tag{3.19}$$

In order to satisfy the no-slip boundary condition, the force $\mathbf{F}(\mathbf{x}, t)$ must be constructed in such a way that the fluid velocity $\mathbf{u}^*(\mathbf{x}, t)$ matches exactly the prescribed boundary velocity $\mathbf{U}(\mathbf{X}_l)$. However, $\mathbf{u}^*(\mathbf{x}, t)$ and $\mathbf{U}(\mathbf{X}_l)$ exist in different reference frames (Eulerian and Lagrangian). Therefore, in order to obtain the boundary velocity $\mathbf{U}^*(\mathbf{X}_l)$ we first need to interpolate $\mathbf{u}^*(\mathbf{x}, t)$ at the boundary nodes. Then, in order to satisfy $\mathbf{U}^*(\mathbf{X}_l) = \mathbf{U}(\mathbf{X}_l)$, a boundary force $\mathbf{F}_l(\mathbf{X}_l)$ must be computed and subsequently spread to the lattice nodes to update the fluid. The overall accuracy and stability of a direct forcing scheme strongly depends on the velocity interpolation and force spreading operations.

3.4.2.1 Multi-Direct Forcing method

The selection of an appropriate direct-forcing method depends on the coupling strategy between the IBM and the LBM. The exact derivation along with a detailed coupling strategy will be presented in Chapter 4. Assuming that Eq. (3.19) is also valid on the IB points and dropping t as the forcing occurs at the same time step, the boundary force $\mathbf{F}_l(\mathbf{X}_l)$ may be simply computed as

$$\mathbf{F}_l(\mathbf{X}_l) = 2\rho\frac{\mathbf{U}(\mathbf{X}_l) - \mathbf{U}^*(\mathbf{X}_l)}{\Delta t}. \tag{3.20}$$

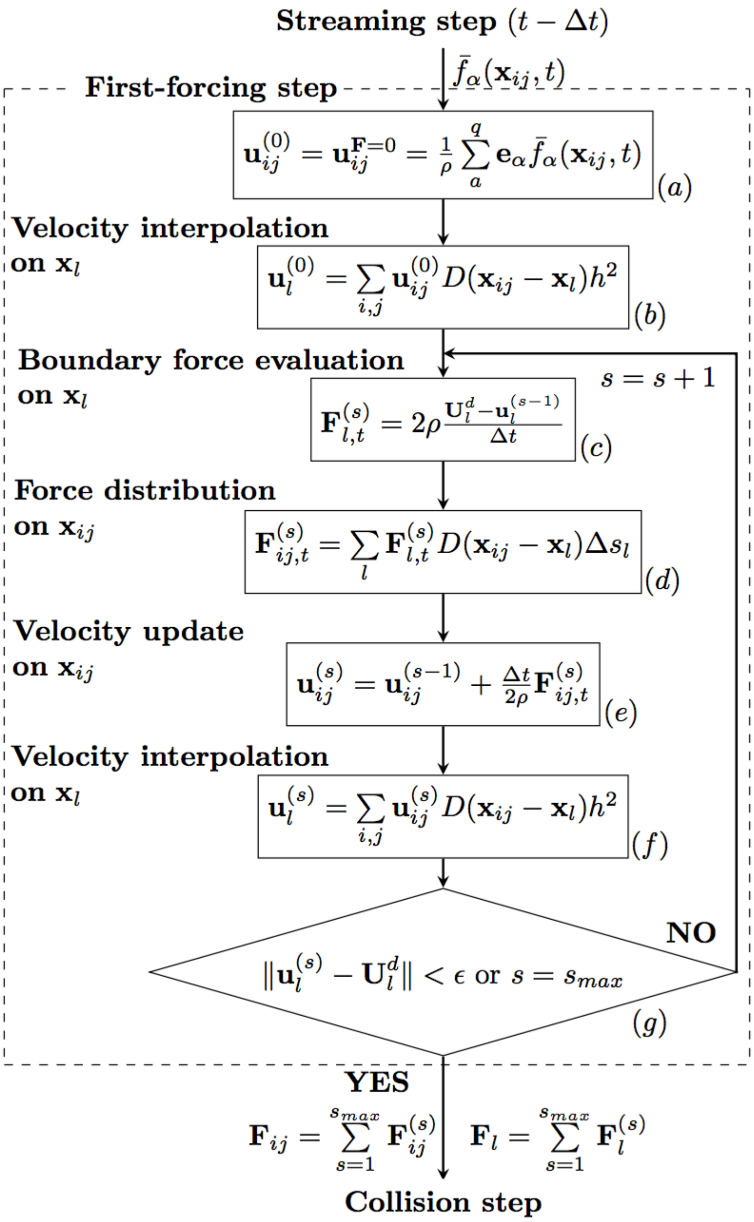
The iterative multi-direct forcing scheme used in this study is similar to the schemes proposed by Luo *et al.* [84], Wang *et al.* [85] and Kang and Hassan [79]. The detailed algorithm is presented in Algorithm 2. The iterations are performed over steps (c) - (f) in order to achieve a sufficiently small difference between the interpolated velocities $\mathbf{U}^*(\mathbf{X}_l)$ and the prescribed velocities $\mathbf{U}^($$\mathbf{X}_l)$. For the sake of clarity, let us define the interpolated velocities as $\mathbf{U}^*(\mathbf{X}_l) = \mathbf{u}_l$ and the desired velocities as $\mathbf{U}(\mathbf{X}_l) = \mathbf{U}_l^d$. The convergence parameter $\epsilon > \left\|\mathbf{u}_l^{(s)} - \mathbf{U}_l^d\right\|$ is somehow arbitrarily set. Unless otherwise specified, ϵ is set at $\epsilon = 10^{-6}$. The implicitness of the MDF algorithm depends on the number of iterations s_{max}.

3.4.2.2 Explicit direct-forcing method

In the explicit direct forcing (EDF) method proposed by Dupuis *et al.* [100], the boundary force $\mathbf{F}_l(\mathbf{X}_l)$ may be simply computed as

$$\mathbf{F}_l(\mathbf{X}_l) = \frac{\mathbf{U}(\mathbf{X}_l) - \mathbf{U}^*(\mathbf{X}_l)}{\Delta t}. \tag{3.21}$$

Algorithm 2 Computational algorithm of the multi direct forcing (MDF)- IBM



Effectively, the EDF is a special case of the multi-direct forcing method where $s = 1$. As shown in 2, using Eq. (3.20) instead of Eq.(3.21) for the force computation, the explicit direct forcing algorithm only includes steps (a)-(e) and no convergence criterion needs to be specified. Various EDF schemes can be found in the literature [126]. However, the EDF approach is not as accurate or stable as the MDF scheme.

3.4.3 Implicit velocity correction method

Apart from the MDF scheme, the above methods do not satisfy the no-slip boundary condition due to insufficient interpolation accuracy. In order to enforce the no-slip boundary condition on a rigid boudary, Wu and Shu [104] proposed the implicit velocity correction IBM scheme. Here, the force density is considered the unknown variable that needs to be computed in such a way that the prescribed BC is satisfied. Using Eq. (3.19), the fluid velocity may be written as

$$\mathbf{u}^*(\mathbf{x},t) = \frac{1}{\rho(\mathbf{x},t)}\sum_{\alpha}^{q}\bar{f}_\alpha(\mathbf{x},t)\mathbf{e}_\alpha + \frac{\mathbf{F}(\mathbf{x},t)\Delta t}{2\rho(\mathbf{x},t)} = \mathbf{u}(\mathbf{x},t) + \delta\mathbf{u}(\mathbf{x},t), \tag{3.22}$$

where $\mathbf{u}(\mathbf{x},t) = \sum_{\alpha}^{q}\bar{f}_\alpha(\mathbf{x},t)\mathbf{e}_\alpha/\rho(\mathbf{x},t)$ is the known velocity given by Eq. (3.17) and $\delta\mathbf{u}(\mathbf{x},t) = \mathbf{F}(\mathbf{x},t)\Delta t/2\rho(\mathbf{x},t)$ is the velocity correction required to satisfy the no-slip BC. The Eulerian velocity correction term $\delta\mathbf{u}(\mathbf{x},t)$ may be expressed in terms of its Lagrangian counterpart $\delta\mathbf{u}_l(\mathbf{X}_l,t)$ using Eq. (3.8) as

$$\delta\mathbf{u}(\mathbf{x},t) = \sum_{l}\delta\mathbf{u}_l(\mathbf{X}_l,t)D(\mathbf{x}_{ij,t} - \mathbf{x}_{l,t})\Delta s_l. \tag{3.23}$$

Using Eqns. (3.22) and (3.23), the reconstructed velocity field $\mathbf{u}^*(\mathbf{x},t)$ must, when interpolated on the boundary nodes, perfectly match the prescribed boundary velocity $\mathbf{U}_l(\mathbf{X}_l)$ as

$$\mathbf{U}_l(\mathbf{X}_l) = \sum_{i,j}\mathbf{u}^*(\mathbf{x}_{ij},t)D(\mathbf{x}_{ij,t} - \mathbf{x}_{l,t})h^2. \tag{3.24}$$

Combining Eqns. (3.22), (3.23) and (3.24) we get the following system of equations

$$\begin{aligned}\mathbf{U}_l(\mathbf{X}_l) =& \sum_{i,j}\mathbf{u}(\mathbf{x}_{ij},t)D(\mathbf{x}_{ij,t} - \mathbf{x}_{l,t})h^2 \\ &+ \sum_{i,j}\sum_{l}\delta\mathbf{u}_l(\mathbf{X}_l,t)D(\mathbf{x}_{ij,t} - \mathbf{x}_{l,t})h^2 D(\mathbf{x}_{ij,t} - \mathbf{x}_{l,t})\Delta s_l.\end{aligned} \tag{3.25}$$

Eq. (3.25) may be re-written in the matrix-vector form $\mathbf{AX} = \mathbf{B}$, where $\mathbf{X} = \delta\mathbf{u}_l(\mathbf{X}_l, t)$ is the L element unknown vector, $\mathbf{A} = \sum\limits_{i,j}\sum\limits_{l} D(\mathbf{x}_{ij,t} - \mathbf{x}_{l,t})h^2 D(\mathbf{x}_{ij,t} - \mathbf{x}_{l,t})\Delta s_l$ is an $L \times L$ matrix whose elements are functions of the Lagrangian nodes and only depend on the kernel function. Finally, the L element vector $\mathbf{B}$ is defined as $\mathbf{B} = \mathbf{U}_l(\mathbf{X}_l) - \sum\limits_{i,j} \mathbf{u}(\mathbf{x}_{ij}, t) D(\mathbf{x}_{ij,t} - \mathbf{x}_{l,t})h^2$. L is the number of Lagrangian points on the boundary. The unknowns X are found by inverting the matrix $\mathbf{A}$, and then spread to the Eulerian grid using Eq. (3.23). Finally, using Eq. (3.22) the fluid velocity is corrected and the force density is computed. Note that for stationary boundary applications, matrix $\mathbf{A}$ only needs to be inverted once since its values depend only on the location of the boundary and the kernel function. However, for moving boundary applications, the inversion of matrix $\mathbf{A}$ at every time step may be computationally expensive when L is large.

3.4.4 Iterative direct forcing method

In order to improve the accuracy of the IBM-LBM scheme, Zhang *et al.* [103] proposed an iterative force correction algorithm based on Cheng's [98] external forcing term for the computation of the boundary forces. The iterative formula for the boundary force correction is

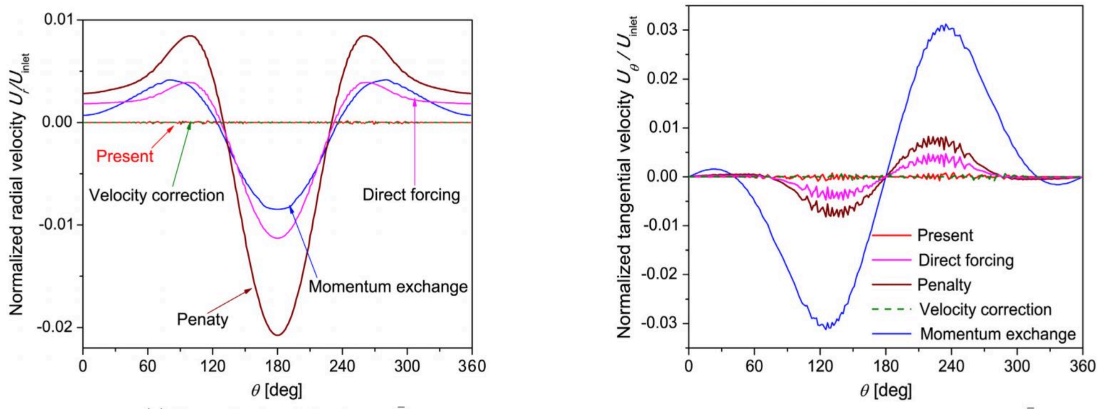
$$\mathbf{F}_l(\mathbf{X}_l) = \mathbf{F}_l^*(\mathbf{X}_l) + 2\rho\frac{\mathbf{U}(\mathbf{X}_l) - \mathbf{U}^*(\mathbf{X}_l)}{\Delta t}, \tag{3.26}$$

which is quite similar to the formula in Eq. (3.20) used for the MDF scheme. One of the major difference of this scheme compared to the MDF is that the fluid velocity does not need to be updated, see step (e) in Fig. 2. The interaction between the boundary and the fluid is performed by splitting the external force into two parts: effects of the present and the next time step. The detailed description of the coupled IBM-LBM algorithm will be presented in Chapter 4.

3.5 Performance properties of the immersed boundary schemes

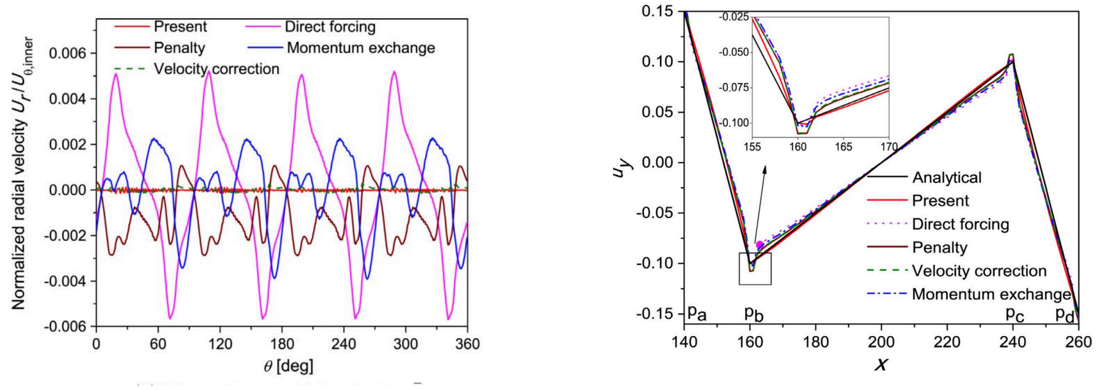
The initial decision on the selection of an accurate and stable IBM method was based on the results reported by Zhang *et al.* [103]. The authors compared the IB schemes presented in this chapter for the flow around a stationary circular cylinder and the flow between two counter rotating cylinders at low Reynolds number flows. Fig. 3.4 shows the normalised radial and tangential velocity components on the surface of the cylinder at $Re = 20$. As observed, the proposed IDF scheme and the implicit velocity correction scheme of Wu and Shu [104] performed significantly better than the other schemes. Both velocity components are accurately computed and spurious oscillations observed on the other schemes are significantly suppressed.

In order to investigate the effect of strong shearing flows on the accuracy of the IBM the authors examined the flow between two counter rotating cylinders. Figures 3.5a and 3.5b shows the normalised radial velocity on the surface of the inner cylinder and the y-component of the velocity along the center horizontal direction for $\nu = 1.0$. Similar observations can be made for both the IDF and the implicit velocity correction schemes. The computed normalised average errors for the two methods were similar and by far smaller than those of the other IB schemes. It is evident that in order to accurately satisfy the no-slip boundary condition and suppress numerical slip on the boundaries either an iterative or an implicit IB scheme should be used. In order to avoid the matrix inversion step in the implicit scheme of Wu and Shu [104], the iterative direct forcing scheme is used in this study. A detailed investigation of the performance of the the IDF scheme coupled with the CLBM is presented in Chapter 6 through various industrial applications. The proposed scheme is also compared with the well established MDF scheme.



(a) **(b)**

Figure 3.4: Comparison of (a) the radial and (b) the tangential velocity components on the surface of a cylinder at $Re = 20$. The results are reported by Zhang *et al.* [103].



(a) **(b)**

Figure 3.5: Comparison of (a) normalised radial velocity on the surface of the inner cylinder and (b) y-component of the velocity along the center horizontal direction. The results are reported by Zhang *et al.* [103].

3.6 Effects of internal mass in the computation of the forces

The effect of the internal mass in the computation of the aerodynamic forces using the immersed boundary method for a moving body is investigated in this section. Only cases with prescribed boundary motions are considered.

3.6.1 Force and Torque acting on a body

The force acting on the immersed body $\mathbf{F}_{tot}$, at time t, can be calculated by the negative sum of the computed Lagrangian forces $\mathbf{F}_l$ as

$$\mathbf{F}_{tot}(t) = -\sum_l \mathbf{F}_l(t)\Delta s_l, \tag{3.27}$$

where Δs_l is the discretisation length of the immersed surface and should be equal for all Lagrangian points l. However, the immersed body receives forces from both the external and the internal fluid, $\mathbf{F}_{IB}$ and $\mathbf{F}_{int}$ respectively. Hence, in order to accurately compute $\mathbf{F}_{IB}$, the total forces $\mathbf{F}_{tot}$ acting on the body must be compensated with the internal forces $\mathbf{F}_{int}$ used to move the internal fluid [86, 87, 81] as

$$\mathbf{F}_{IB}(t) = \mathbf{F}_{tot}(t) + \mathbf{F}_{int}(t). \tag{3.28}$$

The internal forcing term $\mathbf{F}_{int}$, is given by the time derivative of the linear momentum of the internal fluid [86] as

$$\mathbf{F}_{int}(t) = \rho_f \frac{d}{dt} \int\limits_{\mathbf{x}\in\Omega(t)} \mathbf{u}(\mathbf{x},t)d\mathbf{x}, \tag{3.29}$$

where ρ_f is the fluid density. Similarly, the torque acting on the body may be written as a sum of the total and the internal torque

$$\mathbf{T}_{IB}(t) = \mathbf{T}_{tot}(t) + \mathbf{T}_{int}(t), \tag{3.30}$$

$$\mathbf{T}_{tot}(t) = -\sum_l \left[\mathbf{X}_l - \mathbf{X}_c\right] \times \mathbf{F}_l(t)\Delta s_l, \tag{3.31}$$

$$\mathbf{T}_{int}(t) = \rho_f \frac{d}{dt} \int\limits_{\mathbf{x}\in\Omega(t)} \left[\mathbf{X}_l - \mathbf{X}_c\right] \times \mathbf{u}(\mathbf{x},t)d\mathbf{x}. \tag{3.32}$$

$\mathbf{X}_c$ is the center of mass of the body. The two following schemes will be investigated for the computation of the internal mass effects, i.e., the internal force $\mathbf{F}_{int}(t)$ and the internal torque $\mathbf{T}_{int}(t)$.

3.6.2 Rigid body approximation

Assuming a rigid body motion inside the enclosed area $\Omega(t)$, following the work of Feng [88], the integral in Eq. (3.29) can by approximated as $\int_{\mathbf{x}\in\Omega(t)} \mathbf{u}(\mathbf{x},t)d\mathbf{x} = V_b\mathbf{U}_b^p(\mathbf{x},t)$, where $\mathbf{U}_b^p(\mathbf{x},t)$ is the velocity of the internal fluid which satisfies the rigid body motion at all $\mathbf{x} = (x,y) \in \Omega$ and $V_b = M_b/\rho_b$ is the volume of the body. M_b and ρ_b are the mass and density of the rigid body respectively. According to Suzuki *et al.* [86], the equality is theoretically realised since the linear motion of the rigid body is exactly equal to that of the internal mass in spite of actual internal flows. Therefore, the time derivative of the linear momentum of the internal fluid can be approximated as

$$\frac{d}{dt}\int\limits_{\mathbf{x}\in\Omega(t)} \mathbf{u}(\mathbf{x},t)d\mathbf{x} \approx \frac{M_b}{\rho_b}\frac{\mathbf{U}_b^p(t) - \mathbf{U}_b^p(t-\Delta t)}{\Delta t}, \tag{3.33}$$

where $\mathbf{U}_b^p(t)$ is the prescribed body velocity at time t. The approximation is first order accurate in time. Accordingly, the time derivative in Eq. (3.32) be approximated as

$$\frac{d}{dt}\int\limits_{\mathbf{x}\in\Omega(t)} \big[\mathbf{X}_l - \mathbf{X}_c\big]\times\mathbf{u}(\mathbf{x},t)d\mathbf{x} \approx \mathbf{R}(t)\Big[\mathbf{I}_B\frac{\mathbf{\Omega}_b^B(t) - \mathbf{\Omega}_b^B(t-\Delta t)}{\Delta t} + \mathbf{\Omega}_b^B(t)\times(\mathbf{I}_B\mathbf{\Omega}_b^B(t))\Big], \tag{3.34}$$

where $\mathbf{\Omega}_b^B$ is the angular velocity of the body around its centre of mass. In the derivation of Feng [88] and Suzuki *et al.* [86], the reference frame fixed to the body is denoted by B and the inertial frame is denoted by A. Therefore, $\mathbf{R}(t)$ is a rotational matrix transforming the basis vectors from A to B and $\mathbf{I}_B$ is the inertia matrix in B. For a detailed expression of $\mathbf{R}(t)$ and $\mathbf{I}_B$ please refer to Suzuki *et al.* [86].

3.6.3 Lagrangian points approximation

Recently, Suzuki *et al.* approximated the internal forces using internal Lagrangian points moving together with the boundary [86]. A similar approach is used here, focusing more on the identification of the optimum number as well as the distribution of the internal Lagrangian points, in order to optimise both the computational efficiency and the accuracy of the simulation.

1. Compute the surface area A_Ω of the enclosed domain $\Omega(t)$ and equally distribute N_{int} internal Lagrangian points with volume $\Delta V_{int} = 1.0$. N_{int} is the integer value of A_Ω.
2. Interpolate the velocity $\mathbf{U}_{(\mathbf{X}_{int},t)}$ on all internal points using Eq. (3.15).
3. Compute the linear momentum using

$$\mathbf{P}(\mathbf{X}_{int},t) = \Big[\sum_{\mathbf{X}_{int}(t)} \mathbf{U}_{(\mathbf{X}_{int},t)} \Delta V_{int}\Big] S_c. \tag{3.35}$$

4. Compute the angular momentum using

$$\mathbf{L}(\mathbf{X}_{int},t) = \Big[\big[\mathbf{X}_{int} - \mathbf{X}_c\big] \times \sum_{\mathbf{X}_{int}(t)} \mathbf{U}_{(\mathbf{X}_{int},t)} \Delta V_{int}\Big] S_c. \tag{3.36}$$

5. Approximate the internal forces using

$$\mathbf{F}_{int}(t) \approx \Big[\frac{\mathbf{P}(\mathbf{X}_{int},t) - \mathbf{P}(\mathbf{X}_{int},t-\Delta t)}{\Delta t}\Big] S_c. \tag{3.37}$$

6. Approximate the internal torque using

$$\mathbf{T}_{int}(t) \approx \Big[\frac{\mathbf{L}(\mathbf{X}_{int},t) - \mathbf{L}(\mathbf{X}_{int},t-\Delta t)}{\Delta t}\Big] S_c. \tag{3.38}$$

S_c is a scaling factor defined as $S_c = A_\Omega/N_{int}$. Therefore, if half of the internal Lagrangian points are equally distributed for the computation of Eqs. (3.35-3.37), $N_{int} = A_\Omega/2$, then both equations must be multiplied by $S_c = 2$. All internal Lagrangian points follow the prescribed motion of the boundary thus the computational overhead is mainly caused by the interpolation step. This approach differs from the work of Uhlmann [81] where $\mathbf{P}(\mathbf{X}_{int}, t)$ is approximated by summing over internal Eulerian grid points. However, in moving or deformable boundary applications the identification of the internal Eulerian grid points in every time step is computationally complex and expensive.

Chapter 4

Coupling between the LBM and the IBM

The two-way coupling process between the lattice Boltzmann and the immersed boundary method will be presented in this chapter. In this study, the multi direct forcing (MDF) and the iterative direct forcing (IDF) immersed boundary methods have been used. First, the selection of an appropriate discrete force term that is added to the lattice Boltzmann equation will be discussed. That includes a brief review of the most popular forcing schemes in the lattice-Boltzmann framework. In the second part, the effect of the forcing schemes on the overall error and the accuracy of the CLBM is analysed. The novel coupling between the fluid and the solid solvers, as well as the incorporation of the forces in the CLBM used in this study and the proposed IDF-IBM algorithm will be presented.

4.1 Forces in the lattice Boltzmann method

Forces play a significant role in many aerodynamic applications. The most simple example is the force density $\mathbf{F}_g = \rho\mathbf{g}$ originating from the gravitational acceleration g. Adding forces to the LBM is not a trivial task and requires careful consideration of the discrete nature of the fluid solver. The most common way of adding forces in the LBM is by computing the force density and updating the velocity filed prior to the collision step. Once the updated macroscopic variables are known, the collision step is performed. The force density is converted into a discrete source term that is added to the collision kernel. Finally, the populations are propagated and the computation proceeds to the next time step. The form of the force density $\mathbf{F}$ is not given by the LBM and must be computed separately. In this study, the IBM is used to compute $\mathbf{F}$ and model the fluid structure interaction. As shown in Eq. (2.32), half of the force is used to update the velocity prior to computing the equilibrium function in order to achieve second-order space-time accuracy [119].

Let us take for example the SRT-LBM in Eq. (2.19). Once the velocity is updated and the equilibrium distribution computed, the discrete source term may be computed as

$$S_\alpha = \Big(1 - \frac{1}{2\tau}\Big) F_\alpha, \tag{4.1}$$

where F_α is the discrete forcing term. As shown, the relaxation time τ is present in Eq. (4.1). Therefore, when more complex LB collision models are used, e.g., MRT or cascaded, the source term must be computed in the respective moment space. For the sake of clarity let us define the updated "equilibrium" velocity as

$$u_i^{eq} = \frac{1}{\rho} \sum_\alpha^q \bar{f}_\alpha e_{\alpha i} + A \frac{F_i}{\rho}, \tag{4.2}$$

where A is a model dependent parameter. In order to properly construct a forcing scheme both S_α and A must be defined. Table 4.1 summarizes the effect of the forcing schemes that will be investigated in the collision process.

Table 4.1: Effect of different forcing schemes in the collision either through modifying the equilibrium velocity or directly

Scheme	A	S_α
He *et al.* [34]	1/2	$\left(1-\frac{1}{2\tau}\right)\frac{f_\alpha^{eq}}{\rho}\frac{\mathbf{e}_\alpha-\mathbf{u}}{c_s^2}\cdot\mathbf{F}$
Martys *et al.* [95]	1/2	$\left(1-\frac{1}{2\tau}\right)w_\alpha\left(\frac{e_{\alpha i}}{c_s^2}+\frac{(e_{\alpha i}e_{\alpha j}-c_s^2\delta_{ij})u_j}{c_s^4}\right)F_i$
Guo *et al.* [96]	1/2	$\left(1-\frac{1}{2\tau}\right)w_\alpha\left(\frac{\mathbf{e}_\alpha-\mathbf{u}}{c_s^2}+\frac{(\mathbf{e}_\alpha\cdot\mathbf{u})\mathbf{e}_\alpha}{c_s^4}\right)\cdot\mathbf{F}$
Kupershtokh *et al.* [97]	0	$f_\alpha^{eq}(\rho,\mathbf{u}^*+\Delta\mathbf{u})-f_\alpha^{eq}(\rho,\mathbf{u}^*)$
Cheng and Li [98]	0	$\frac{1}{2}\left[F_\alpha(\mathbf{x}+\mathbf{e}_\alpha\delta t,t+\delta t)+F_\alpha(\mathbf{x},t)\right],$ $F_\alpha=w_\alpha\left\{3\mathbf{F}\cdot\left[(\mathbf{e}_\alpha-\mathbf{u})+3(\mathbf{e}_\alpha\mathbf{u})\mathbf{e}_\alpha\right]\right\}$

4.1.1 The method of Guo *et al.*

Following the work of Ladd and Verberg [89], Guo *et al.* [96] investigated the lattice effects in the presence of an external force. The authors concluded that the forcing and source terms take the forms of Eq.(4.6) and Eq. (4.1) respectively. In addition, assuming $A = 1/2$ removes undesired derivatives in the continuity and momentum equation.

4.1.2 The method of Kupershtokh *et al.*

Kupershtokh *et al.* [97] proposed the so-called exact difference method (EDF). The scheme is based on the kinetic theory and the forcing term is expressed in terms of the equilibrium functions f_α^{eq} alone in order to avoid shifting the populations f_α in velocity space. Effectively, the force density is expressed as a velocity correction term as $\Delta\mathbf{u} = \mathbf{F}\delta t/\rho$. Assuming that the presence of a force does not change the equilibrium state, the source term may be given as

$$S_\alpha = f_\alpha^{eq}(\rho, \mathbf{u}^* + \Delta\mathbf{u}) - f_\alpha^{eq}(\rho, \mathbf{u}^*), \tag{4.3}$$

where $\mathbf{u}^*$ is the "unforced" velocity. It can be seen that the effect of the forcing term is independent of the relaxation time τ.

4.1.3 The method of He *et al.*

In Chapter 2, the strategy for incorporating force terms in the cascaded collision model has been discussed. In the work of Premnath [94], the strategy proposed by He *et al.* [34] has been used. Starting from the continuous Boltzmann equation with the SRT collision model, He *et al.* added an external body force term which can depend on both space and time as

$$\frac{\partial f}{\partial t} + \boldsymbol{\xi} \cdot \nabla f + \mathbf{F} \cdot \nabla_\xi f = \frac{1}{\tau}\big(f - f^{\mathscr{M}}\big). \tag{4.4}$$

As shown in Chapter 2, the derivative $\mathbf{F} \cdot \nabla_\xi$ may be simplified using Eq. (2.8). For the SRT LBM, Eq. (2.9) takes the following form

$$F_\alpha = \frac{f_\alpha^{eq}}{\rho} \frac{\mathbf{e}_\alpha - \mathbf{u}}{c_s^2} \cdot \mathbf{F}, \tag{4.5}$$

and the equilibrium velocity in Eq. (4.2) is modified with $A = 1/2$.

4.1.4 The method of Martys *et al.*

In the work of Martys *et al.* [95] the derivative $\mathbf{F} \cdot \nabla_\xi$ in Eq. (4.4) has been evaluated using an expansion in the Hermite polynomial series. Truncating the expanded forcing term up to second velocity order the discrete forcing term may be written as

$$F_\alpha = w_\alpha\Big(\frac{e_{\alpha i}}{c_s^2} + \frac{(e_{\alpha i}e_{\alpha j} - c_s^2\delta_{ij})u_j}{c_s^4}\Big)F_i. \tag{4.6}$$

Similar to the work of He *et al.* [34], $A = 1/2$. In addition, it can be shown that Eq. (4.6) satisfies the first three velocity moments as $\sum_\alpha F_\alpha = 0$, $\sum_\alpha F_\alpha e_{\alpha,i} = F_i$ and $\sum_\alpha F_\alpha e_{\alpha,i}e_{\alpha,j} = F_i u_j + F_j u_i$.

4.1.5 The method of Cheng and Li

In order to handle both time and space dependencies of the external forces, Cheng and Li [98] derived a scheme where the effects of both the current and the next time step are incorporated in the source term without modifying the velocity term. It has been shown, both numerically and theoretically, that the method is second-order accurate within the incompressible limit of the LBM [127]. Similar to the EDM [97] the effect of the forcing term is independent of the relaxation time τ and thus can be directly applied to any collision model. Effectively, the source term S_α may be expressed as

$$S_\alpha = \frac{1}{2}\big[F_\alpha(\mathbf{x} + \mathbf{e}_\alpha\delta t, t + \delta t) + F_\alpha(\mathbf{x}, t)\big], \tag{4.7}$$

where the discrete forcing term F_α is given by

$$F_\alpha = w_\alpha\big\{A + 3\mathbf{B}\cdot\big[(\mathbf{e}_\alpha - \mathbf{u}) + 3(\mathbf{e}_\alpha\mathbf{u})\mathbf{e}_\alpha\big]\big\}. \tag{4.8}$$

According to Cheng and Li *et al.* [98], A is the source term in the continuity equation and $\mathbf{B}$ is the forcing term in the continuity equation. Thus, we let $A = 0$ and $\mathbf{B} = \mathbf{F}$. In order to "predict" the effect of the forcing term on the next time step $F_\alpha(\mathbf{x} + \mathbf{e}_\alpha\delta t, t + \delta t)$, an iterative procedure must be used to solve the implicit Eq. (4.7).

4.2 Error from forcing schemes

Prior to selecting an appropriate forcing scheme for the LBM it is important to understand how it affects the overall accuracy and stability of the method. In the cascaded LBM framework, the effect on the forcing on the higher order moments must be considered. First, let us write the LBM with a generic collision operator and forcing term as

$$\bar{f}_\alpha(\mathbf{x}+\mathbf{e}_\alpha\Delta t, t+\Delta t) = \bar{f}_\alpha + \Omega_\alpha + S_\alpha. \tag{4.9}$$

Taking the zeroth, first and second moments of S_α we get $\sum_\alpha S_\alpha = 0$, $\sum_\alpha \mathbf{e}_\alpha S_\alpha = \mathbf{F}$ and $\sum_\alpha \mathbf{e}_\alpha \mathbf{e}_\alpha S_\alpha = \mathbf{C}$ respectively. As in [128], $\mathbf{C}$ is a general second order moment that should be equal to $\mathbf{C} = (\mathbf{Fu}+\mathbf{uF})$ for the continuous LBM. However, as shown by the authors, in the discrete LBM additional discretisation errors occur and should be cancelled by the selected forcing scheme. In the work of Lycett-Brown and Luo [128] the introduced error may be expressed as

$$\mathbf{E} = \tau(\mathbf{Fu}+\mathbf{uF}) - \tau\mathbf{C} + \Big(\tau - \frac{1}{4}\Big)\frac{\mathbf{FF}}{\rho}. \tag{4.10}$$

All the forcing schemes presented in this chapter successfully cancel the first term in the error equation reducing the error to

$$\mathbf{E} = \frac{\mathbf{FF}}{\rho}\Big(\tau - \frac{1}{4}\Big). \tag{4.11}$$

However, the second order moment $\mathbf{C}$ should include additional terms that cancel the third term in Eq. (4.10). Introducing the above expressions for the different forcing schemes, the respective errors are $\mathbf{E}^{EDM} = -\frac{1}{4}\frac{\mathbf{FF}}{\rho}$ and $\mathbf{E}^{Guo} = \mathbf{E}^{Martys} = 0$.

The exact difference method of Kupershtokh *et al.* [97] introduce some additional terms in the recovered macroscopic equations. The effects of those terms on the numerical performance of the EDM have not yet been investigated.

In the scheme of He *et al.* [34], the discretisation error in Eq. (4.11) does not cancel and depends on the relaxation time τ. However, in their derivation the use of the trapezoidal rule in the discretisation of the force integral results in an explicit scheme that can be written in the form of Eq. (4.9). As stated by Lycett-Brown and Luo [128] the force term of He *et al.* [34] only differs to the scheme of Guo *et al.* [96] at third order.

Kang and Hassan [79] show that the forcing scheme of Cheng and Li [98] is effectively the implicit version of the Guo scheme, $\mathbf{E}^{Cheng} = 0$. In the proposed scheme, the authors specifically redefined the forcing term in order to satisfy $\mathbf{C} = \sum_\alpha \mathbf{e}_\alpha \mathbf{e}_\alpha S_\alpha = \mathbf{Fu} + \mathbf{uF} + c_s^2 \delta \sum_\alpha S_\alpha$, where δ is the kronecker delta. According to the authors, the previous forcing schemes are accurate only for steady and unsteady body forces as the derivative of the force moments are zero, cancelling any discretisation error terms in the Navier-Stokes equations. In their method, the implicitness of the proposed forcing term satisfies the moment constrains and accurately recovers the NS equations.

For the interested reader, a detailed investigation of different forcing schemes on the accuracy and performance of the cascaded LBM can be found in [129].

4.3 Forces in the cascaded lattice Boltzmann method

As shown in Chapter 2, collisions in the CLBM are performed in the central moment space following a cascaded manner. Two approaches of incorporating external forces in the cascaded collision operator are investigated in this study.

First, the forcing term of He *et al.* [34] is used. As shown in Table 4.1 the source term of He *et al.* depends on the relaxation time τ. Therefore, in order to consistently incorporate the forcing term in the CLBM the method of Premnath [94] is followed. The detailed form of the updated collision and force terms can be found in Chapter 2. The velocity is modified according to Eq. (2.32). In order to accurately prescribe the BCs on the immersed body, the multi-direct forcing IBM is coupled with the fluid solver. The coupling algorithm is shown in section 4.3.

The forcing scheme of Cheng and Li *et al.* [98] is used in the second approach. Since the force term does not depend on the relaxation parameter τ, the discrete source term may be directly added to the LBM equation as

$$f_\alpha(\mathbf{x}+\mathbf{e}_\alpha\Delta t, t+\Delta t) = f_\alpha(\mathbf{x}, t)+(K\cdot\hat{\mathbf{k}})_\alpha+\frac{1}{2}\left[F_\alpha(\mathbf{x}+\mathbf{e}_\alpha\delta t, t+\delta t)+F_\alpha(\mathbf{x}, t)\right]. \quad (4.12)$$

Since an iterative scheme will be used for the computation of $F_\alpha(\mathbf{x}+\mathbf{e}_\alpha\delta t, t+\delta t)$, the transformation of the populations in Eq. (2.24) is no longer necessary thus we drop the bar over f_α. We follow the same method for the construction of the collision operator. All force terms in Eqns (2.33) - (2.38) are removed and the source term S_α is given by Eq. (4.7) and Eq. (4.8). Finally, the velocity term in Eq. (2.32) does not need to be modified. The iterative direct forcing IBM will be used for the computation of the force density on the boundary nodes and the discrete term $F_\alpha(\mathbf{x}+\mathbf{e}_\alpha\delta t, t+\delta t)$.

4.4 Coupling algorithms

In this section, the coupling algorithms used in this study are presented. In both methods the force is applied in two steps during a single time step as shown in Algorithms 3 and 4. Note that in Algorithm 3, the collision kernel $\hat{\mathbf{k}}_\alpha$ has the

Algorithm 3 Multi direct forcing, split-forcing LMB algorithm

```
First forcing step: $\rho(\mathbf{x},t)\mathbf{u}(\mathbf{x},t)) = \sum_\alpha \mathbf{e}_\alpha \bar{f}_\alpha(\mathbf{x},t) + \frac{\delta t}{2}\mathbf{F}(\mathbf{x},t)$.
Collision step: $\bar{f}^{p1}_\alpha(\mathbf{x},t) = \bar{f}_\alpha(\mathbf{x},t) + (K \cdot \hat{\mathbf{k}})_\alpha$.
Second forcing step: $\bar{f}^{p2}_\alpha(\mathbf{x},t) = \bar{f}^{p1}_\alpha(\mathbf{x},t) + \delta t F_\alpha(\mathbf{x},t)$.
Streaming step: $\bar{f}_\alpha(\mathbf{x}+\mathbf{e}_\alpha\Delta t, t+\Delta t) = \bar{f}^{p2}_\alpha(\mathbf{x},t)$.
```

Algorithm 4 Iterative direct forcing, split-forcing LMB algorithm

```
First forcing step: $f^{'}_\alpha(\mathbf{x},t) = f_\alpha(\mathbf{x},t) + \frac{\delta t}{2}F_\alpha(\mathbf{x},t)$.
Collision step: $f^{p1}_\alpha(\mathbf{x},t) = f^{'}_\alpha(\mathbf{x},t) + (K \cdot \hat{\mathbf{k}}^{'})_\alpha$.
Second forcing step: $f^{p2}_\alpha(\mathbf{x},t) = f^{p1}_\alpha(\mathbf{x},t) + \frac{\delta t}{2}F_\alpha(\mathbf{x},t)$.
Streaming step: $f_\alpha(\mathbf{x}+\mathbf{e}_\alpha\Delta t, t+\Delta t) = f^{p2}_\alpha(\mathbf{x},t)$.
```

form of Eqns. (2.30) - (2.35), whereas in Algorithm 4 the unforced moments $\hat{\mathbf{k}}^{'}_\alpha$ are used.

4.4.1 The multi-direct forcing immersed boundary cascaded lattice Boltzmann - MDF-CLBM

In the multi direct forcing algorithm, an iterative procedure is employed for the computation of $\mathbf{F}(\mathbf{x},t)$ in the first forcing step in order to satisfy the no slip boundary condition on the boundary points. First, let us consider the unforced particle distribution functions under no external force at $(\mathbf{x}, t+\Delta t)$. Eq. (2.32) can be written as

$$\rho(\mathbf{x},t+\Delta t)\mathbf{u}^{\mathbf{F}=0}(\mathbf{x},t+\Delta t) = \sum_a^q \mathbf{e}_\alpha \bar{f}_\alpha(\mathbf{x},t+\Delta t). \tag{4.13}$$

Assuming that the no slip boundary condition is satisfied by a desired velocity $\mathbf{U}^d$ at the next time step, Eq. (2.32) may be written as

$$\rho(\mathbf{x},t+\Delta t)\mathbf{U}^d = \sum_a^q \mathbf{e}_\alpha \bar{f}_\alpha(\mathbf{x},t+\Delta t) + \frac{\Delta t}{2}\mathbf{F}(\mathbf{x},t+\Delta t). \tag{4.14}$$

Substracting Eq. (4.13) from Eq. (4.14), the desired force can be obtained as

$$\mathbf{F}(\mathbf{x}, t+\Delta t) = 2\rho(\mathbf{x}, t+\Delta t)\frac{\mathbf{U}^d - \mathbf{u}^{\mathbf{F}=0}(\mathbf{x}, t+\Delta t)}{\Delta t}. \tag{4.15}$$

Eq. (4.15) is assumed to be also valid at the Lagrangian points at the next time step, i.e. $\mathbf{x} \mapsto \mathbf{X}_{\mathscr{L}}$. The detailed description of the algorithm may be found in 3.4.2.1 and in Fig. 2. Once the force density on the boundary points has been computed and the velocity is updated, the post collision distributions may be explicitly computed as

$$f_0^p = f_0 + 4(-\hat{k}_3 + \hat{k}_8) + F_0, \tag{4.16a}$$

$$f_1^p = f_1 - \hat{k}_3 + \hat{k}_4 + 2(\hat{k}_7 - \hat{k}_8) + S_1, \tag{4.16b}$$

$$f_2^p = f_2 - \hat{k}_3 - \hat{k}_4 + 2(\hat{k}_6 - \hat{k}_8) + S_2, \tag{4.16c}$$

$$f_3^p = f_3 - \hat{k}_3 + \hat{k}_4 - 2(\hat{k}_7 + \hat{k}_8) + S_3, \tag{4.16d}$$

$$f_4^p = f_4 - \hat{k}_3 - \hat{k}_4 - 2(\hat{k}_6 + \hat{k}_8) + S_4, \tag{4.16e}$$

$$f_5^p = f_5 + 2\hat{k}_3 + \hat{k}_5 - \hat{k}_6 - \hat{k}_7 + \hat{k}_8 + S_5, \tag{4.16f}$$

$$f_6^p = f_6 + 2\hat{k}_3 - \hat{k}_5 - \hat{k}_6 + \hat{k}_7 + \hat{k}_8 + S_6, \tag{4.16g}$$

$$f_7^p = f_7 + 2\hat{k}_3 + \hat{k}_5 + \hat{k}_6 + \hat{k}_7 + \hat{k}_8 + S_7, \tag{4.16h}$$

$$f_8^p = f_8 + 2\hat{k}_3 - \hat{k}_5 + \hat{k}_6 - \hat{k}_7 + \hat{k}_8 + S_8. \tag{4.16i}$$

The exact form of $\hat{k}_\alpha$ and S_α can be found in Chapter 2. Finally the post-collision states are streamed along the discrete directions $\mathbf{e}_\alpha$ and proceed to the next time step.

4.4.2 The iterative direct forcing immersed boundary cascaded lattice Boltzmann - IDF-CLBM

In order to derive an iterative expression for the computation of the force density on the boundary nodes and the force term $F_\alpha(\mathbf{x} + \mathbf{e}_\alpha \delta t, t + \delta t)$ in Eq. (4.12) at

$t+\delta t$, we introduce an intermediate forcing term $F^*_\alpha(\mathbf{x}+\mathbf{e}_\alpha\delta t, t+\delta t)$. Introducing the intermediate terms in the transport equation Eq. (4.12) then reads

$$f^*_\alpha(\mathbf{x}+\mathbf{e}_\alpha\Delta t, t+\Delta t) = f^d_\alpha(\mathbf{x}, t)+(K\cdot\hat{\mathbf{k}})_\alpha+\frac{1}{2}\left[F^*_\alpha(\mathbf{x}+\mathbf{e}_\alpha\delta t, t+\delta t)+F^d_\alpha(\mathbf{x}, t)\right], \quad (4.17)$$

where the superscript d denotes the desired values obtained at the current time-step in order to satisfy the no-slip boundary condition and $f^*_a(\mathbf{x}+\mathbf{e}_a\delta t, t+\delta t)$ are the intermediate distributions at $t+\delta t$. Following the algebraic manipulations as in Zhang *et al.*, the desired external force may be rewritten as

$$\begin{aligned}\mathbf{f}^d(\mathbf{x}+\mathbf{e}_a\delta t, t+\delta t) &= \mathbf{f}^*(\mathbf{x}+\mathbf{e}_a\delta t, t+\delta t)\\ &\quad + 2\frac{\mathbf{u}^d(\mathbf{x}+\mathbf{e}_a\delta t, t+\delta t)-\mathbf{u}^*(\mathbf{x}+\mathbf{e}_a\delta t, t+\delta t)}{\delta t}\end{aligned} \quad (4.18)$$

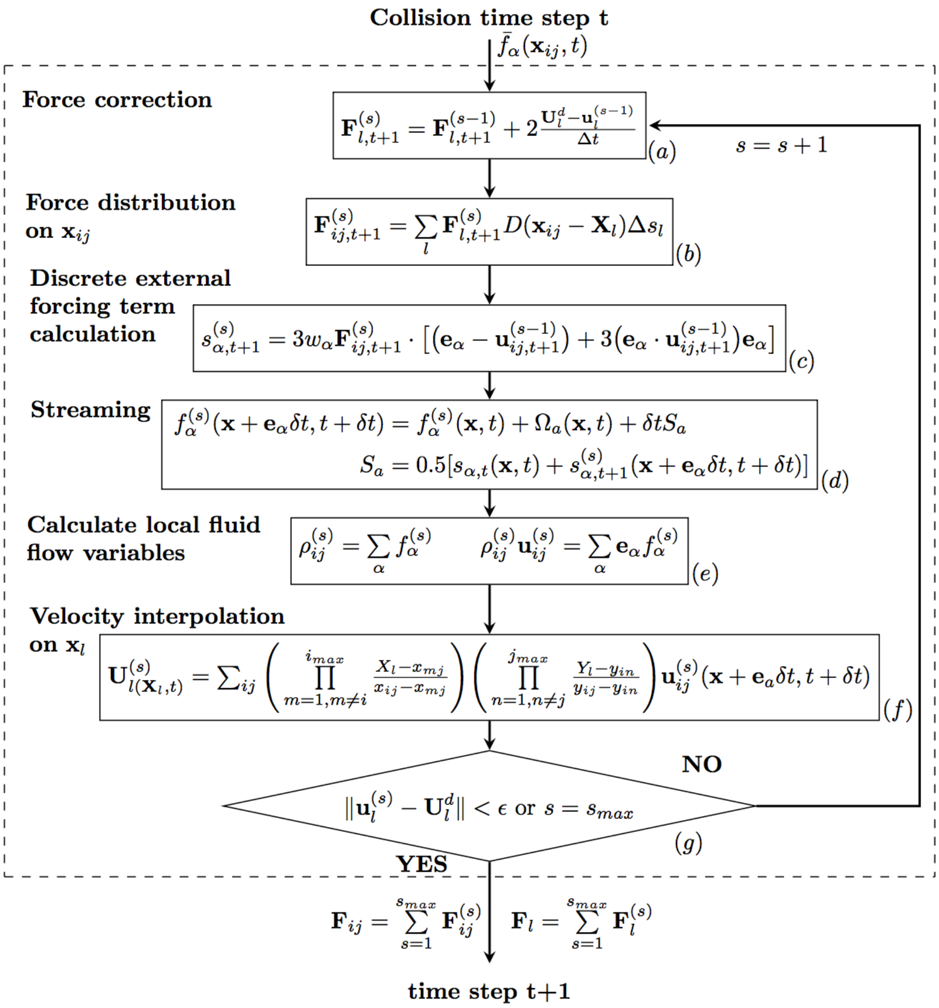
In order for the no-slip boundary condition to be satisfied, the immersed boundary velocity must be equal to the fluid velocity at the Lagrangian points. Under the assumption that Eq. (4.18) is also valid at the Lagrangian points, we obtain the following iterative formula for the boundary force correction.

$$\mathbf{F}^d(\mathbf{X}_l, t) = \mathbf{F}^*(\mathbf{X}_l, t) + 2\frac{\mathbf{U}^d(\mathbf{X}_l, t)-\mathbf{U}^*(\mathbf{X}_l, t)}{\Delta t}. \quad (4.19)$$

$\mathbf{U}^d(\mathbf{X}_l, t)$ is the desired prescribed boundary velocity, i.e. equals to zero in the case of a stationary boundary. Thus, when $\mathbf{U}^*(\mathbf{X}_l, t)$; the intermediate velocity obtained from interpolations, is equal to $\mathbf{U}^d(\mathbf{X}_l, t)$, the desired force must be equal to the intermediate force at all boundary points. Accordingly, the desired force at all the neighboring Eulerian grid points must be equal to the respective intermediate force, $\mathbf{f}^d(\mathbf{x}+\mathbf{e}_a\Delta t, t+\Delta t) = \mathbf{f}^*(\mathbf{x}+\mathbf{e}_a\Delta t, t+\Delta t)$.

The detailed computational algorithm for the iterative direct forcing scheme is shown in Algorithm 5. As shown in algorithm 4 the forcing is applied in two steps, before and after streaming of the populations. The IFD scheme in

Algorithm 5 Computational algorithm of the iterative force correction (IDF)-IBM



Algorithm 5 is applied on the third step of Algorithm 4, after collision has been performed. Finally, similar to the MDF coupling scheme, the post-collision states are streamed along the discrete directions $\mathbf{e}_\alpha$ and proceed to the next time step.

Chapter 5

Hybrid expanded higher-order immersed boundary method

A novel immersed boundary method with a pressure correction algorithm is presented. The proposed expanded higher-order immersed boundary method addresses two major drawbacks of the conventional IBM.

First, an expanded velocity profile scheme will be derived in Section 5.2 in order to compensate for the discontinuities caused by the gradient of the velocity across the boundary. Second, a numerical method derived from the Navier-Stokes equation in order to correct the pressure distribution across the boundary will be presented in detail in Section 5.3. The performance of the proposed hybrid scheme will be demonstrated in Chapter 8 by investigations of the flow around stationary and moving boundaries.

5.1 Major drawbacks of the the conventional IBM

The significance of the interpolation accuracy on the overall accuracy of an IBM, as well as the selection of a proper kernel function have been discussed in Chapter 3. The IDF scheme presented in Chapter 4, improves the global accuracy of the solution by incorporating the effects of both the current and next time step in the source term without modifying the velocity term.

However, similar to other IB schemes, the force spreading operation used to enforce the prescribed boundary conditions creates a discontinuity of the velocity gradient across the boundary [130]. That leads to a local, first-order accurate solution in space. Inspired by the method of Suzuki and Inamuro [130], we present an improved scheme where the boundary layer is smoothly reconstructed inside the body domain in order to suppress the velocity gradient discontinuities.

Compared to other existing IBM-LBM coupling schemes, the accuracy of the pressure computation is improved in the IDF-CLBM [55, 56] by directly incorporating the forces in the discrete populations. However, additional actions must be considered. Suzuki and Inamuro [130] show that the pressure, though not expected, is indirectly affected by the improvement of the accuracy on the computed velocity. In the ghost fluid lattice-Boltzmann method (GF-LBM) proposed by Tiwari and Vanka [131], Neumann boundary conditions have been imposed for the pressure on the boundary by taking a zero normal gradient condition at the walls as $\partial_n p = 0$. However, this assumption is oversimplified, especially in moving boundary applications and under the presence of IB forces. In order to improve the solution accuracy of the pressure field near the boundary, a novel hybrid scheme, derived directly from the Navier-Stokes equations, will be derived in this study.

5.2 Expanded higher-order immersed boundary method

In this section we propose a novel hybrid higher order IBM that addresses the issues discussed previously. A coherent description of the numerical method is presented. We follow the principles of the iterative direct forcing (IDF) immersed boundary scheme, as shown in Chapter 4. In order to smoothly expand the velocity profile inside the body domain, an ensemble of fluid and ghost Lagrangian nodes is used. The effect of the additional Lagrangian nodes in the computation of the force density around the boundary will be presented in subsection 5.2.1. The proposed algorithm, coupled with the CLBM is described in detail in subsection 5.2.2. Unless otherwise specified the notations and definitions are the same as the ones presented in Chapter 3. Lagrangian and Eulerian quantities are denoted with capital and lower case letters respectively.

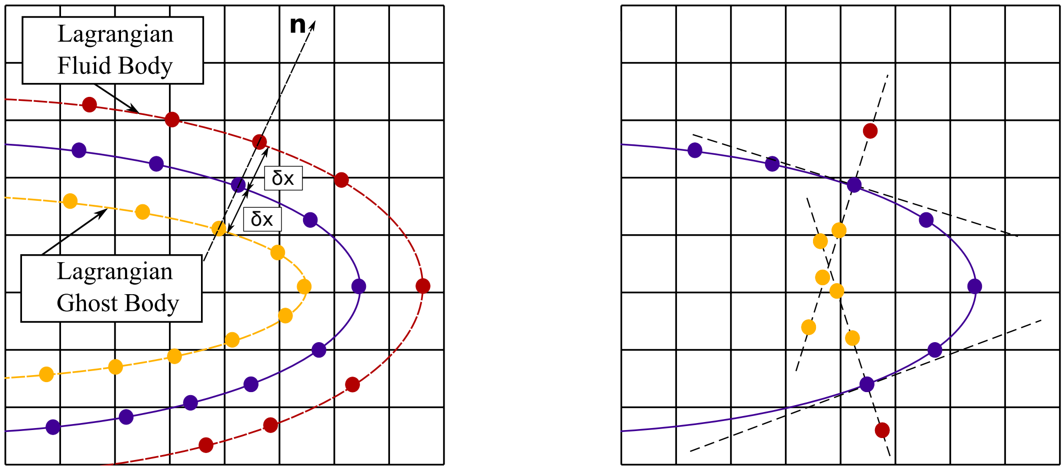
5.2.1 Force density computation

Consider an ensemble of $\mathscr{N}$ Lagrangian points $\mathbf{X}^b_{\mathscr{L}}(t+\Delta t)$ that discretize the boundary, with respective velocities $\mathbf{u}^b_{\mathscr{L}}(t+\Delta t)$. Taking the normal unit vector at each boundary point and pointing outwards, a set of Lagrangian fluid nodes $\mathbf{X}^{f_k}_{\mathscr{L}}(t+\Delta t)$ for $k = 1, 2, \cdots, N_f$ as well as, a set of Lagrangian ghost nodes inside the body domain $\mathbf{X}^{g_m}_{\mathscr{L}}(t+\Delta t)$, for $m = 1, 2, \cdots, N_g$ are defined as

$$\mathbf{X}^{g_m}_{\mathscr{L}} = \mathbf{X}^b_{\mathscr{L}} - m(\Delta x)\mathbf{n}_{\mathscr{L}} \tag{5.1}$$

$$\mathbf{X}^{f_k}_{\mathscr{L}} = \mathbf{X}^b_{\mathscr{L}} + k(\Delta x)\mathbf{n}_{\mathscr{L}}, \tag{5.2}$$

where the superscripts *m and k* refer to the number of Lagrangian ghost and fluid node sets respectively. We define the Lagrangian fluid and ghost nodes as points



(a) (b)

Figure 5.1: (a): Locations of fluid $\mathbf{X}^{f_k}_{\mathscr{L}}$ and ghost $\mathbf{X}^{g_m}_{\mathscr{L}}$ Lagrangian nodes, (b): Forbidden placement of ghost nodes.

that reside inside the fluid domain outside and inside the boundary respectively. Fig. 5.1a shows a specific configuration with $m = k = 1$. Note that if either the lattice spacing or the curvature of the boundary are large, two consecutive line segments of ghost nodes could be intersected, Fig. 5.1b. That would cause the velocity at the intersection to be multivalued and thus should be avoided. In the case of a moving boundary, the locations of $\mathbf{X}^{f_k}_{\mathscr{L}}$ and $\mathbf{X}^{g_m}_{\mathscr{L}}$ are updated at the start of each time iteration. The Lagrangian interpolation polynomials are used for the computation of the temporal velocities $\mathbf{U}^*(\mathbf{X}^{f_k}_{\mathscr{L}}, t + \delta t)$ at the fluid points $\mathbf{X}^{f_k}_{\mathscr{L}}$ as

$$\mathbf{U}^*(\mathbf{X}^{f_k}_{\mathscr{L}}) = \sum_{ij} \left(\prod_{m=1,m\neq i}^{i_{max}} \frac{X^{f_k}_{\mathscr{L}} - x_{mj}}{x_{ij} - x_{mj}} \right) \left(\prod_{n=1,n\neq j}^{j_{max}} \frac{Y^{f_k}_{\mathscr{L}} - y_{in}}{y_{ij} - y_{in}} \right) \mathbf{u}_{ij}(\mathbf{x} + \mathbf{e}_a \delta t, t + \delta t), \tag{5.3}$$

where i, j is the kernel range for the interpolation. Eq. (5.3) is also used for the computation of the intermediate velocities at the ghost nodes $\mathbf{U}^*(\mathbf{X}^{g_m}_{\mathscr{L}}, t + \delta t)$.

In order to impose the prescribed Dirichlet boundary conditions, the interpolated velocities on the fluid nodes must be expanded inside the boundary. In the work of Suzuki and Inamuro [130], the expansion is performed by a first-order linear extrapolation scheme. The desired velocity on the ghost nodes $\mathbf{U}^{g_m}_{\mathscr{L}} = \mathbf{U}(\mathbf{X}^{g_m}_{\mathscr{L}})$ may be computed as

$$\mathbf{U}^{g_m}_{\mathscr{L}} = \mathbf{U}^{b}_{\mathscr{L}} + \left(\frac{\mathbf{X}^{g_m}_{\mathscr{L}} - \mathbf{X}^{b}_{\mathscr{L}}}{\mathbf{X}^{f_1}_{\mathscr{L}} - \mathbf{X}^{b}_{\mathscr{L}}}\right)\big(\mathbf{U}^{f_1}_{\mathscr{L}} - \mathbf{U}^{b}_{\mathscr{L}}\big) = \mathbf{U}^{b}_{\mathscr{L}} - m\big(\mathbf{U}^{f_1}_{\mathscr{L}} - \mathbf{U}^{b}_{\mathscr{L}}\big), \tag{5.4}$$

where $\mathbf{U}^{b}_{\mathscr{L}} = \mathbf{U}(\mathbf{X}^{b}_{\mathscr{L}})$ is the prescribed boundary velocity and $\mathbf{U}^{f_1,*}_{\mathscr{L}} = \mathbf{U}^{*}(\mathbf{X}^{f_1}_{\mathscr{L}})$ is the interpolated velocity on the fluid nodes $\mathbf{X}^{f_1}_{\mathscr{L}}$. As shown in Fig. 5.2a, only one fluid node is required. It may be seen from Eq. (5.4) that in the case of a stationary boundary the velocity on the ghost nodes $\mathbf{U}^{g_m}_{\mathscr{L}}$ is simply given by $\mathbf{U}^{g_1}_{\mathscr{L}} = -\mathbf{U}^{f_1}_{\mathscr{L}}$, $\mathbf{U}^{g_2}_{\mathscr{L}} = -2\mathbf{U}^{f_1}_{\mathscr{L}}$ and $\mathbf{U}^{g_3}_{\mathscr{L}} = -3\mathbf{U}^{f_1}_{\mathscr{L}}$. The main drawback of this method is that at high shear flows, the expanded velocity profile inside the boundary gets non-physical, high values, leading to unstable flow conditions near the boundary. More sophisticated extrapolation schemes may be used in order to overcome this issue.

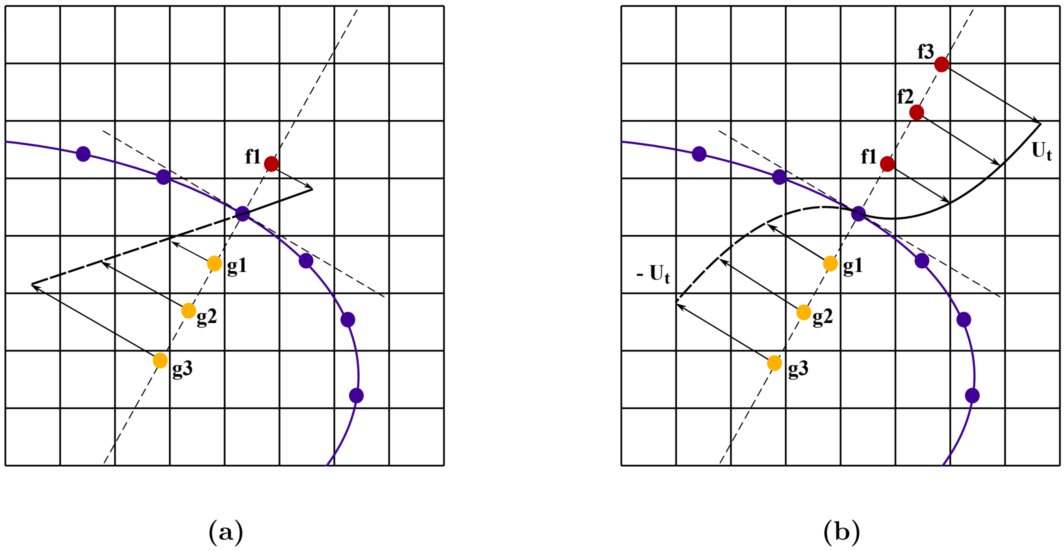
In the present study, we choose an equal number of Lagrangian fluid and ghost nodes, $m = k$. The idea is to smoothly expand the boundary layer profile inside the boundary. Therefore, the desired ghost node velocities $\mathbf{U}(\mathbf{X}^{g_m}_{\mathscr{L}})$ are computed as

$$\mathbf{U}(\mathbf{X}^{g_m}_{\mathscr{L}}) = 2\mathbf{U}(\mathbf{X}^{b}_{\mathscr{L}}) - \mathbf{U}(\mathbf{X}^{f_m}_{\mathscr{L}}). \tag{5.5}$$

The proposed scheme is shown in Fig. 5.2b. Following the strategy shown in Chapter 4, the body forces that must be applied both at the boundary and the ghost nodes are given by

$$\mathbf{F}^{(s)}(\mathbf{X}^{b}_{\mathscr{L}}, t+\Delta t) = \mathbf{F}^{(s-1)}(\mathbf{X}^{b}_{\mathscr{L}} t+\Delta t) + 2\left(\frac{\mathbf{U}^{b}_{\mathscr{L}} - \mathbf{U}^{*}(\mathbf{X}^{b}_{\mathscr{L}}, t+\Delta t)}{\Delta t}\right), \tag{5.6}$$

$$\mathbf{F}^{(s)}(\mathbf{X}^{g_m}_{\mathscr{L}}, t+\Delta t) = \mathbf{F}^{(s-1)}(\mathbf{X}^{g_m}_{\mathscr{L}}, t+\Delta t) + 2\left(\frac{\mathbf{U}^{g_m}_{\mathscr{L}} - \mathbf{U}^{*}(\mathbf{X}^{g_m}_{\mathscr{L}}, t+\Delta t)}{\Delta t}\right), \tag{5.7}$$



(a) (b)

Figure 5.2: (a): Extrapolation of the velocity inside the ghost domain, (b): Full reconstruction of the boundary layer profile inside the ghost domain. The Lagrangian fluid nodes are shown in red, whereas the Lagrangian ghost nodes are shown in yellow. The distance between consecutive ghost nodes is $\delta x = 1.0$.

where s denotes the iteration step in the IDF algorithm. The intermediate velocities $\mathbf{U}^*(\mathbf{X}^b_{\mathscr{L}}, t+\Delta t)$ and $\mathbf{U}^*(\mathbf{X}^{g_m}_{\mathscr{L}}, t+\Delta t)$ may be computed using Eq. (5.3). Finally, the force densities are spread into the Eurelian grid using

$$\begin{aligned}\mathbf{f}^{(s)}(\mathbf{x}, t+\Delta t) = &\sum_{\mathscr{L}=1}^{\mathscr{N}} \mathbf{F}^{(s)}(\mathbf{X}^b_{\mathscr{L}}, t+\Delta t)\delta_h(\mathbf{x}-\mathbf{X}^b_{\mathscr{L}})\Delta S^b_{\mathscr{L}} \\ &+ \sum_{m=1}^{N_g}\sum_{\mathscr{L}=1}^{\mathscr{N}} \mathbf{F}^{(s)}(\mathbf{X}^{g_m}_{\mathscr{L}}, t+\Delta t)\delta_h(\mathbf{x}-\mathbf{X}^{g_m}_{\mathscr{L}})\Delta S^{g_m}_{\mathscr{L}}, \qquad (5.8)\end{aligned}$$

where $\Delta S^b_{\mathscr{L}}$ and $\Delta S^{g_m}_{\mathscr{L}}$ are the arc lengths of the boundary and ghost elements respectively.

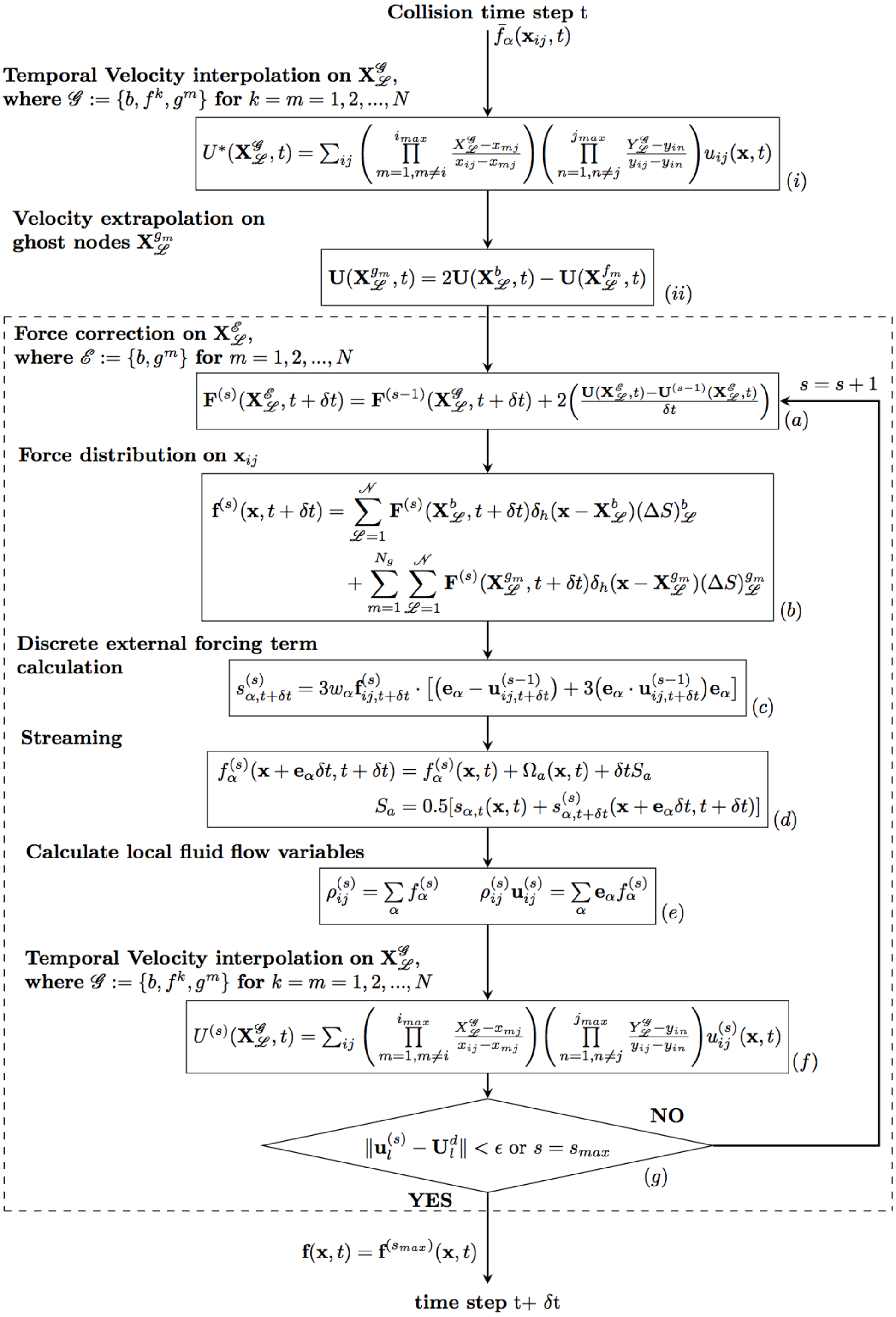
Algorithm 6 Computational algorithm for the expanded higher-order iterative force correction IBM

```
1: Update the position of all Lagrangian bodies: $\mathbf{X}^{b}_{\mathscr{L}}$, $\mathbf{X}^{g_m}_{\mathscr{L}}$ and $\mathbf{X}^{f_m}_{\mathscr{L}}$.
2: Interpolate velocity on $\mathbf{X}^{b}_{\mathscr{L}}$, $\mathbf{X}^{g_m}_{\mathscr{L}}$ and $\mathbf{X}^{f_m}_{\mathscr{L}}$ using Eq. (5.3) and
    a: Compute the temporal velocities on the fluid nodes, $\mathbf{U}(\mathbf{X}^{f_m}_{\mathscr{L}})$.
    b: Compute the intermediate velocities on the body and ghost nodes, $\mathbf{U}^{*}(\mathbf{X}^{b}_{\mathscr{L}})$ and $\mathbf{U}^{*}(\mathbf{X}^{g_m}_{\mathscr{L}})$.
3: Compute the desired ghost node velocities $\mathbf{U}^{d}(\mathbf{X}^{g_m}_{\mathscr{L}})$ using Eq. (5.5).
4: Compute body forces on boundary and ghost nodes, , $\mathbf{F}(\mathbf{X}^{b}_{\mathscr{L}})$ and $\mathbf{F}(\mathbf{X}^{g_m}_{\mathscr{L}})$, using Eqns. ((5.6) - (5.7)).
5: Distribute the force densities on the Eulerian grid using Eq. (5.8).
6: Compute the discrete lattice forcing term $S_\alpha$ using Eq. (4.8).
7: Stream populations using Eq. (4.12).
8: Compute the macroscopic density $\rho$ and velocity $\mathbf{u}$ using Eq. (2.32).
9: Compute the intermediate velocities on the body and ghost nodes, $\mathbf{U}^{*}(\mathbf{X}^{b}_{\mathscr{L}})$ and $\mathbf{U}^{*}(\mathbf{X}^{g_m}_{\mathscr{L}})$
10: Repeat steps (4) - (9) until the convergence criteria are satisfied.
```

5.2.2 Proposed Algorithm

The proposed computational algorithm for the expanded higher-order iterative force correction IBM is explained in Algorithms 6 and 7. The computational algorithm is similar to the one shown in Chapter 4. Effectively, if $m = 0$ the expanded higher order scheme falls back to the IDF-CLBM coupling scheme. The proposed scheme may be expressed as a series of IDF schemes, each of which computes a unique force density distribution, in order to prescribe the desired boundary conditions on the respective Lagrangian boundary ($\mathbf{X}^{b}_{\mathscr{L}}$ and $\mathbf{X}^{g_m}_{\mathscr{L}}$). As shown in Algorithm 7, Step b, the coupling is simply done using Eq. (5.8).

Algorithm 7 Expanded computational algorithm for the expanded higher-order iterative force correction IBM



5.3 Pressure correction algorithm

The second part of the proposed hybrid scheme involves the derivation of a numerical method from the Navier-Stokes equation, in order to prescribe accurate pressure boundary conditions on the boundary. In immersed boundary methods, the fluid flow near the boundary is characterized by the magnitude and nature of the forces exerted by the boundary on the fluid. It will be shown that the Neumann, zero normal gradient boundary condition for the pressure $\partial_n p = 0$ is an oversimplification in the IBM framework. Based on the expanded higher order IBM formulation, we derive a novel pressure boundary condition where both the external forces and the fluid velocity near the boundary are considered. A set of Lagrangian fluid and ghost nodes will be used for the implementation of the BCs.

In subsection 5.3.1, the derivation of the novel pressure BC will be presented. Finally, the implementation of the algorithm in the LBM framework will be explained in subsection 5.3.2.

5.3.1 Derivation from the Navier-Stokes equations - Assumptions

The 2D Navier - Stokes equations in the x and y axes may be written as

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\frac{\partial^2 u}{\partial x^2} + \nu\frac{\partial^2 u}{\partial y^2} + f_x, \tag{5.9a}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial y} + \nu\frac{\partial^2 v}{\partial x^2} + \nu\frac{\partial^2 v}{\partial y^2} + f_y. \tag{5.9b}$$

Before proceeding with the derivation, further assumptions must be made. Assuming $\frac{\partial u}{\partial x} \ll \frac{\partial u}{\partial y}$ and $\frac{\partial v}{\partial x} \ll \frac{\partial v}{\partial y}$, Eq. (5.9a) may be written as

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\frac{\partial^2 u}{\partial y^2} + f_x, \tag{5.10a}$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=-\frac{1}{\rho}\frac{\partial p}{\partial y}+\nu\frac{\partial^2 v}{\partial y^2}+f_y. \tag{5.10b}$$

The next assumptions refer to the order of magnitude of the variables in Eq. (5.9a). It occurs that $t \sim T$, $u \sim U$, $v \sim U\delta/L$, $x \sim L$, $y \sim \delta$, $\nu \sim \delta^2$ and $f \sim ML/T^2$, where U is the freestream velocity, T is a characteristic time length, δ is the boundary layer thickness as shown in Fig. 5.3, L is a characteristic length and M is a characteristic mass. It may be realised that both the convective and the viscous terms in Eqns. (5.10a) are not dominant as $u\frac{\partial v}{\partial x} \sim v\frac{\partial v}{\partial y} \sim U^2\delta/L^2$ and $\nu\frac{\partial^2 v}{\partial y^2} \sim U\delta/L$ and thus may be neglected. In this scheme, only pressure variations in the direction normal to the wall, $\mathbf{n}$, will be considered. Therefore, using Eq. (5.10a), the proposed pressure boundary condition may be written as

$$\frac{\partial u_{\mathbf{n}}}{\partial t}=-\frac{1}{\rho}\frac{\partial p}{\partial n}+f_{\mathbf{n}}, \tag{5.11}$$

where $u_{\mathbf{n}}$ and $f_{\mathbf{n}}$ are the projected on the normal direction velocity and force respectively. Note that Eq. (5.11) is only consistent at the fluid region where u and v are sufficiently small. Therefore, only the first, closest to the wall Lagrangian fluid node f_1 will be used for the implementation of the boundary condition, as shown in Fig. 5.4a.

Using the first fluid and ghost nodes, Eq. (5.11) may be written in a discrete form as

$$\frac{u^{f_1}_{\mathbf{n},t}-u^{f_1}_{\mathbf{n},t-1}}{\delta t}=-\frac{1}{\rho^{f_1}}\frac{p^{f_1}_t-p^{g_1,d}_t}{2\alpha\delta x}+f^{f_1}_{\mathbf{n},t}, \tag{5.12}$$

where $p^{g_1,d}_t=p^{g_1}_t+p^{g_1,*}_t$ is the desired pressure on the ghost node g_1, $p^{g_1,*}_t$ is a pressure correction term and α is an adjusting term position of the ghost nodes. The value of α will be discussed in the following section. Eq. (5.12) is first order accurate in time and second order accurate in space. In the LBM, the pressure is related to the density as $p=\rho c_s^2$, where $c_s=1/\sqrt{3}$ is the speed of sound on the D2Q9 lattice model. The pressure correction term $\rho^{g_1,*}_t$ on the ghost nodes g_1

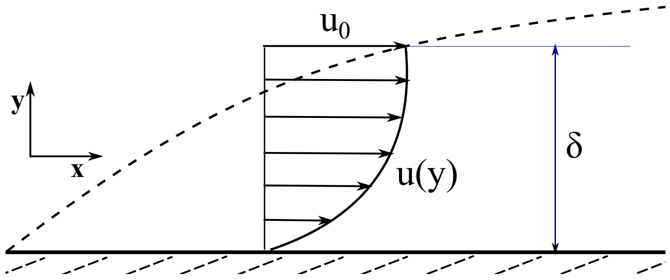


Figure 5.3: Fluid flow and boundary layer thickness δ over a flat wall.

may now be computed as

$$\rho_t^{g_1,*} = \Delta\rho_t^{f_1/g_1} + 6\rho_t^{f_1}\alpha\delta x\Big(\frac{u_{\mathbf{n},t}^{f_1} - u_{\mathbf{n},t-1}^{f_1}}{\delta t} - f_{\mathbf{n},t}^{f_1}\Big), \tag{5.13}$$

where $\Delta\rho_t^{f_1/g_1} = \rho_t^{f_1} - \rho_t^{g_1}$. The Lagrangian variables $\rho_t^{f_1}$, $\rho_t^{g_1}$, $u_{\mathbf{n},t}^{f_1}$, $u_{\mathbf{n},t-1}^{f_1}$ and $f_{\mathbf{n},t}^{f_1}$ are interpolated on the markers using the Lagrangian interpolation polynomials as in Eq. (5.3).

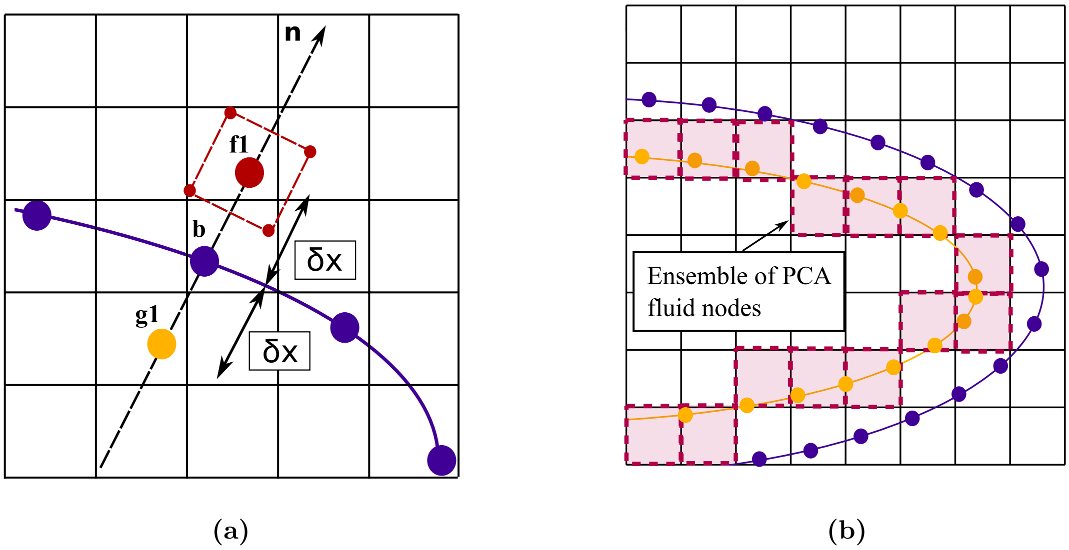
The proposed algorithm may be applied to more general cases where the convective and viscous terms in Eq. 5.10a could not be neglected. That would require computation of the respective derivatives in Eq. 5.10a on the normal and tangential to the boundary directions and would impose additional computational overhead. Note that the implementation of the algorithm would not change and any additional terms would only be added to Eq. 5.13.

5.3.2 Pressure correction algorithm

First, let us rewrite Eq. (5.13) in Lagrangian formulation as

$$\rho_t^*(\mathbf{X}_{\mathscr{L}}^{g_1}) = \Delta\rho_t(\mathbf{X}_{\mathscr{L}}^{f_1/g_1}) + 6\rho_t(\mathbf{X}_{\mathscr{L}}^{f_1})\alpha\delta x\Big(\frac{U_{\mathbf{n},t}(\mathbf{X}_{\mathscr{L}}^{f_1}) - U_{\mathbf{n},t-1}(\mathbf{X}_{\mathscr{L}}^{f_1})}{\delta t} - F_{\mathbf{n},t}(\mathbf{X}_{\mathscr{L}}^{f_1})\Big), \tag{5.14}$$

where $\Delta\rho_t(\mathbf{X}_{\mathscr{L}}^{f_1/g_1}) = \rho_t(\mathbf{X}_{\mathscr{L}}^{f_1}) - \rho_t(\mathbf{X}_{\mathscr{L}}^{g_1})$. In order to correct the fluid pressure inside the immersed body, the pressure correction term $\rho^*(\mathbf{X}_{\mathscr{L}}^{g_1}, t)$ computed by



(a) (b)

Figure 5.4: (a): Fluid and ghost node placement for the pressure correction algorithm (PCA) with $\alpha = 1.0$, (b): Ensemble of Eulerian nodes where the pressure correction algorithm is applied.

Eq. (5.14) on the Lagrangian ghost nodes g_1 must be spread onto the Eulerian grid. To that end, using the spreading kernel of the IBM the Eulerian pressure correction may be computed as

$$\rho^*(\mathbf{x}, t) = \sum_{\mathscr{L}=1}^{\mathscr{N}} \rho^*(\mathbf{X}_{\mathscr{L}}^{g_1}, t)\delta_h(\mathbf{x} - \mathbf{X}_{\mathscr{L}}^{g_1})(\Delta S)_{\mathscr{L}}^{g_1}. \tag{5.15}$$

As shown in Fig. 5.4a, in order to avoid spreading $\rho^*(\mathbf{X}_{\mathscr{L}}^{g_1}, t)$ onto any Eulerian nodes outside the boundary, the 2-point hat function $\phi_1(r)$ will be used in Eq. (5.15) as

$$\phi_1(r) = \begin{cases} 1 - |r| & \text{if } |r| \leq 1.0 \\ 0 & \text{if } 1.0 \leq |r| \end{cases}. \tag{5.16}$$

Accordingly, the term α is set at $\alpha = 1.2$. Finally, the desired lattice densities that satisfy the boundary condition in Eq. (5.13) are computed on the ensemble shown in Fig. 5.4b as

$$\rho^d(\mathbf{x},t) = \rho(\mathbf{x},t) + \rho^*(\mathbf{x},t). \tag{5.17}$$

The final stage of the pressure correction algorithm (PCA) includes the reconstruction of the populations $f_\alpha = f_\alpha^{(eq)} + f_\alpha^{(neq)}$ on the nodes where the density has been updated. Following the same principle, the non-equilibrium parts of the distribution functions are interpolated on the Lagrangian fluid and ghost nodes using the Lagrangian interpolation polynomials as

$$f^{(neq)}_{\mathscr{L},\alpha}(\mathbf{X}^{\mathscr{A}}_{\mathscr{L}},t) = \sum_{ij}\left(\prod_{m=1,m\neq i}^{i_{max}} \frac{X^{\mathscr{A}}_{\mathscr{L}} - x_{mj}}{x_{ij} - x_{mj}}\right)\left(\prod_{n=1,n\neq j}^{j_{max}} \frac{Y^{\mathscr{A}}_{\mathscr{L}} - y_{in}}{y_{ij} - y_{in}}\right) f^{(neq)}_\alpha(\mathbf{x},t), \tag{5.18}$$

where $\mathscr{A} := \{f_1, g_1\}$. The desired non-equilibrium distributions $f^{(neq),d}_{\mathscr{L},\alpha}(\mathbf{X}^{g_1}_{\mathscr{L}},t)$ on the ghost nodes g_1 are obtained using a first order extrapolation scheme as $f^{(neq),d}_{\mathscr{L},\alpha}(\mathbf{X}^{g_1}_{\mathscr{L}},t) = f^{(neq)}_{\mathscr{L},\alpha}(\mathbf{X}^{f_1}_{\mathscr{L}},t)$. The non-equilibrium correction distributions are then computed as

$$\Delta f^{(neq)}_{\mathscr{L},\alpha}(\mathbf{X}^{g_1}_{\mathscr{L}},t) = f^{(neq),d}_{\mathscr{L},\alpha}(\mathbf{X}^{g_1}_{\mathscr{L}},t) - f^{(neq)}_{\mathscr{L},\alpha}(\mathbf{X}^{g_1}_{\mathscr{L}},t). \tag{5.19}$$

Finally, the desired non-equilibrium functions on the Eulerian nodes may be computed as

$$f^{(neq),d}_\alpha(\mathbf{x},t) = f^{(neq)}_\alpha(\mathbf{x},t) + \Delta f^{(neq)}_\alpha(\mathbf{x},t), \tag{5.20}$$

where

$$\Delta f^{(neq)}_\alpha(\mathbf{x},t) = \sum_{\mathscr{L}=1}^{\mathscr{N}} \Delta f^{(neq)}_{\mathscr{L},\alpha}(\mathbf{X}^{g_1}_{\mathscr{L}},t)\delta_h(\mathbf{x}-\mathbf{X}^{g_1}_{\mathscr{L}})(\Delta S)^{g_1}_{\mathscr{L}}. \tag{5.21}$$

The populations f_α can now be reconstructed as

$$f_\alpha(\mathbf{x},t) = f^{(eq)}_\alpha\big(\mathbf{u}(\mathbf{x},t)\rho^d(\mathbf{x},t)\big) + f^{(neq),d}_\alpha(\mathbf{x},t). \tag{5.22}$$

The proposed computational algorithm for the hybrid pressure correction scheme is described in 9 and compactly shown in 8. The interpolation stencil $\mathscr{K}^{g_1}$ is

Algorithm 8 Computational algorithm for hybrid pressure correction scheme.

```
1: Compute the macroscopic variables $\rho$, $\mathbf{u}$ and the equilibrium $f^{(eq)}$ and non-equilibrium $f^{(neq)}$ parts of the populations.
2: Compute $\rho(\mathbf{X}_{\mathscr{L}}^{f_1},t), \rho(\mathbf{X}_{\mathscr{L}}^{g_1},t), \mathbf{U}(\mathbf{X}_{\mathscr{L}}^{f_1},t), \mathbf{F}(\mathbf{X}_{\mathscr{L}}^{f_1},t), f_{\mathscr{L},\alpha}^{(neq)}(\mathbf{X}_{\mathscr{L}}^{f_1},t), f_{\mathscr{L},\alpha}^{(neq)}(\mathbf{X}_{\mathscr{L}}^{g_1},t)$ using the Lagrange interpolation polynomials.
3: Project $\mathbf{U}(\mathbf{X}_{\mathscr{L}}^{f_1},t), \mathbf{U}(\mathbf{X}_{\mathscr{L}}^{f_1},t-1), \mathbf{F}(\mathbf{X}_{\mathscr{L}}^{f_1},t)$ on the normal directions $\mathbf{n}^{f_1}$.
4: Compute the density correction and non-equilibrium terms on $\mathbf{X}_{\mathscr{L}}^{g_1}$ using Eq. (5.13) and Eq. (5.19).
5: Distribute the density correction and the non-equilibrium correction on the ensemble of Eulerian nodes using Eq. (5.15) and (5.21).
6: Update the density and the non-equilibrium parts on the ensemble using Eq. (5.17) and Eq. (5.20).
7: Reconstruct the local populations $f_\alpha^{'}$ on the ensemble using Eq. (5.22).
```

defined as $\mathscr{K}^r = \left(\prod_{m=1,m\neq i}^{i_{max}} \frac{X_{\mathscr{L}}^r - x_{mj}}{x_{ij}-x_{mj}}\right)\left(\prod_{n=1,n\neq j}^{j_{max}} \frac{Y_{\mathscr{L}}^r - y_{in}}{y_{ij}-y_{in}}\right)$. The brackets in steps (a) and (b) refer to a mathematical notation, where each included variable separately undergoes the respective operation.

The effect of the proposed scheme on the mass conservation is investigated by examining the rate of change of the total mass in the surface and the net inflow of mass as

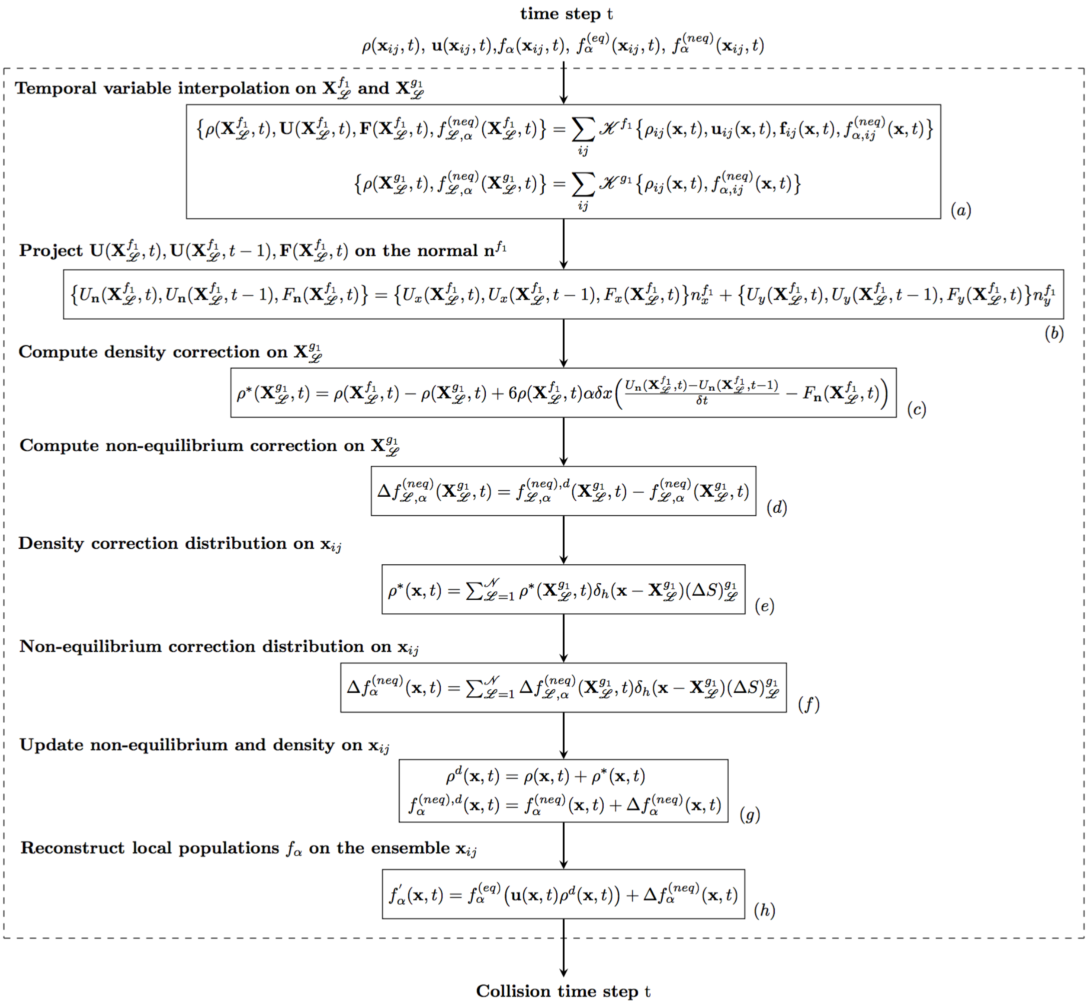
$$\frac{d}{dt}\iiint \rho dV = -\oint \rho \mathbf{u}\cdot\mathbf{n} dA. \tag{5.23}$$

Reducing Eq. 5.23 to 2 dimensions we get

$$\frac{d}{dt}\iint \rho dA = -\oint \rho \mathbf{u}\cdot\mathbf{n} ds. \tag{5.24}$$

In order to examine the effect of the proposed scheme, Eq. 5.24 has been computed on a stationary circular cylinder at steady flow conditions at $Re = 40$. The rate of change of the total mass in the immersed surface was $\frac{d}{dt}\iint \rho dA = 0$

Algorithm 9 Expanded hybrid pressure correction algorithm



between consecutive time steps. The net mass inflow before and after the immersed boundary method was $\oint \rho\mathbf{u}\cdot\mathbf{n}ds = 0.00036$ and $\oint \rho\mathbf{u}\cdot\mathbf{n}ds = 0.0000086$ respectively. The effect of this small mass flux across the boundary must be further examined.

Chapter 6

Computational Results: Flows around stationary objects

Most realistic fluid flow problems are characterised by high Reynolds numbers and complex boundaries. Over the last ten years, immersed boundary methods that are able to cope with realistic geometries have been applied to Lattice-Boltzmann (LB) methods. These methods, however, have normally been applied to low Reynolds number problems. Here we present a novel coupling between an iterative force-correction immersed boundary (Zhang *et al.*, 2016) and a multi-domain cascaded LB method. The iterative force-correction immersed boundary method has been selected due to the improved accuracy of the computation, while the cascaded LB formulation is used due to its superior stability at high Reynolds numbers. The coupling is shown to improve both the stability and numerical accuracy of the solution. The resulting solver has been applied to viscous flow (up to a Reynolds number of 100000) passed a NACA-0012 airfoil at a 10 degree angle of attack. Good agreement with results obtained using a body-fitted Navier-Stokes solver has been obtained. The formulation provides a straight forward and efficient method for modelling realistic geometries including both stationary and moving boundaries.

Effectively, the difference in modelling stationary and moving boundaries lies in the treatment of the internal mass and the prescription of the velocity boundary conditions on the IBM. In moving boundary applications, the motion of the boundary should be prescribed in such a way that certain conditions are satisfied. This is further discussed in Chapter 7.

6.1 2D Taylor Green Decaying vortex - IBM

Lattice Boltzmann methods are second-order accurate in time and space. However, since the interpolation steps in the immersed boundary method are only first order accurate in space, the effect on the global accuracy of the solution must be investigated. In order to evaluate the accuracy of the proposed IDF-CLBM coupling scheme, the two-dimensional unsteady and fully periodic Taylor-Green vortex flow in a square box is investigated. A circle with diameter $D = 0.5L$ is immersed at the center of square domain $L \times L$ as shown in Fig. 6.1. The cylinder is placed in the domain in order to investigate the accuracy of the present IBM scheme by imposing the analytical velocity values as velocity boundary conditions on the immersed surface. We aim at investigating the effect of the IBM on the formal 2^{nd} order accuracy of the CLBM. In practice, ideally, the presence of the cylinder and in turn the coupling of the IBM with the LBM, should produce the same results as if the fluid flow was solved only with the LBM.

The analytical solutions for the velocity and pressure are of the form:

$$\mathbf{u}(\mathbf{x},t) = U_0 \left\{ \begin{array}{c} -\sqrt{k_y/k_x}\cos(k_x x)\sin(k_y y) \\ \\ \sqrt{k_y/k_x}\sin(k_x x)\cos(k_y y) \end{array} \right\} e^{-t/t_d}, \tag{6.1}$$

$$p(\mathbf{x},t) = p_0 - \rho U_0^2/4\big[k_y/k_x \cos(2k_x x) + k_x/k_y \cos(2k_y y)\big]e^{-2t/t_d}, \tag{6.2}$$

where U_0 is the initial velocity, $k_x = k_y = 2\pi/L$ are the wave vector $\mathbf{k}$ components and $t_d = [\nu(k_x^2 + k_y^2)]^{-1}$ is the decay time of the vortex. The flow field is initialised using Eqs. (6.1-6.2). The velocity boundary conditions on the immersed body are prescribed using Eq. (6.1). Periodic boundary conditions have been used for the domain boundaries. The global error of the velocities is evaluated at time $t^* = tD/U_\infty = 1$ using the following L_2 norm error:

$$L_2^{error} \equiv \sqrt{\Big(\sum (u_x^c - u_x^a)^2 + (u_y^c - u_y^a)^2\Big)/N}, \tag{6.3}$$

where the summation is over the total number of grid nodes N and the superscripts, a and c, refer to the analytical and computational values respectively. Fig. 6.1 shows the velocity magnitude and vector plots at $t^* = 1$, for $Re = 10$ and $\tau = 0.65$. We investigate the convergence rate of the present scheme for two cases. For the definition of the apparent order of convergence the formula $p = \ln\big((L_2^{i+1} - L_2^{i})/(L_2^{i} - L_2^{i-1})\big)/\ln(r)$ and a constant grid refinement ratio $r = 2$ are used. The superscripts in the L_2 terms denote the refinement levels

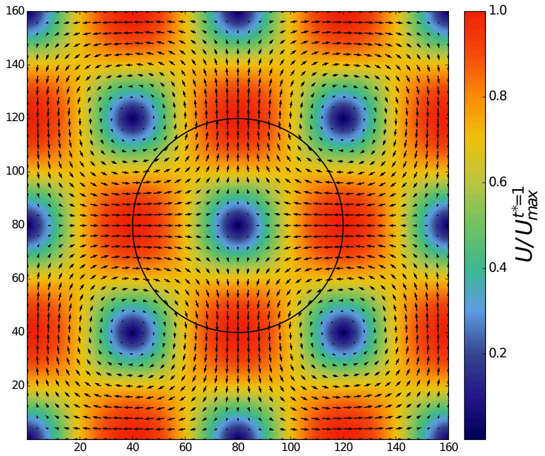
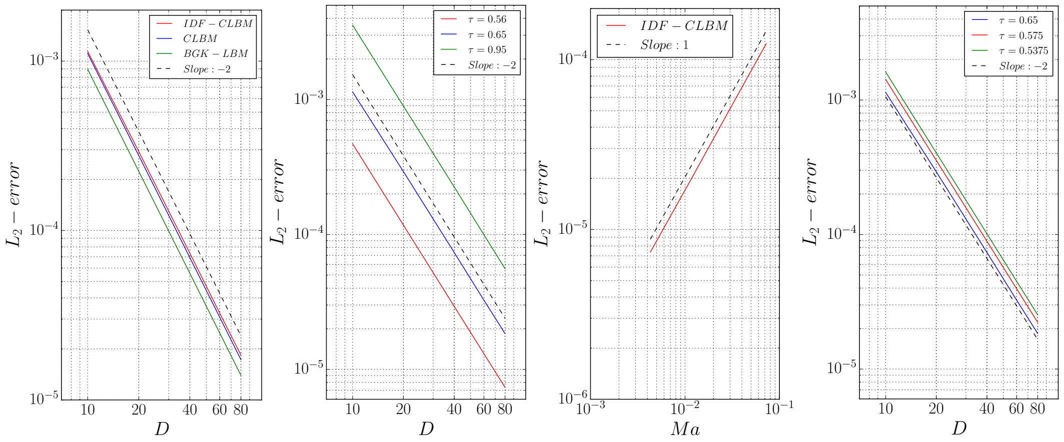


Figure 6.1: Velocity magnitude and vector plots at $t^* = 1$, resulting from the IDF-CLBM scheme.

($D = 10, 20, 40, 80$). In case 1, the Reynolds number is set to $Re = U_0D/\nu = 10$ and the relaxation time is set to be $\tau = 0.65$ as in [104]. Fig. 6.2a shows the global L_2-error versus the number of grid points along the cylinder. It is shown that the global second order accuracy of the CLBM is not significantly affected by the IBM. The apparent order of convergence is $p = 1.992$ for the CLBM, $p = 1.998$ for the BGK-LBM, $p = 1.986$ for the IDF-CLBM between the refining region ($D = 20 - 80$) and $p = 1.958$ for the IDF-CLBM between the refining region ($D = 10 - 40$). The L_2-error is slightly lower for the BGK LBM and in agreement with the values reported in [79]. Wu and Shu [104] reported a convergence rate of 1.9 and Kang and Hassan [79] a convergence rate of 1.98. Fig. 6.2b shows that as the fluid viscosity and the Mach number, $Ma = U_0/c_s$, decrease ($\tau = 0.95, 0.65, 0.56$) the magnitude of the error L_2-decreases while the convergence rate is not affected. The convergence rate of the L_2-error with respect to the Mach number is first order ($p \approx 1$) , as shown in Fig. 6.2c. In case 2, the flow at $Re = 10, 20, 40$ is investigated. The fluid velocity U_0 is kept constant for $Re = 10, 20, 40$ and the relaxation parameter is set to $\tau = 0.65, 0.575, 0.5375$. Fig. 6.2d shows that the overall accuracy of the method is not affected by the relaxation parameter. However, as the fluid viscosity decreases, a small increase in the magnitude of the L_2-error is observed.

6.2 Circular Cylinder

The first validation example is the simulation of flows past a circular cylinder. In this extensively studied, both numerically and experimentally, problem, the flow behaviour changes according to the Reynolds number, which is defined as $Re = u_\infty D/\nu$. D is the diameter of the cylinder, u_∞ is the freestream velocity and ν is the kinematic viscosity. In order to avoid the transition range, $Re = 150-300$, of the flow to 3D [132] [133], $Re = 20, 40, 100$ and 150 are examined. In the present method, the external forces can be computed directly from the boundary forces



(a) **(b)** **(c)** **(d)**

Figure 6.2: (a) Overall accuracy of the IDF-CLBM scheme. (b) Effect of the relaxation parameter on the overall accuracy for $Re = 10$. (c) Effect of the Mach number on the overall accuracy for $Re = 10$. (d) Effect of the relaxation parameter on the overall accuracy for constant fluid velocity and $Re = 10, 20, 40$.

obtained from the IDF scheme. For the lift force, the spanwise component $F_{l,y}$, of the boundary force is used

$$F_L = -\sum_l F_{l,y} \Delta s_l, \tag{6.4}$$

whereas the streamwise component $F_{l,x}$ of the boundary force is used for the computation of the drag force.

$$F_D = -\sum_l F_{l,x} \Delta s_l. \tag{6.5}$$

The summation is performed over all Lagrangian points. Δs_l is the discretisation length of the immersed surface and should be equal for all points l. Unless otherwise specified, the forcing point are uniformly distributed with $\Delta s_l = 1.0$.

Using Eqs. (6.4 - 6.5), the lift and drag coefficients are defined as

$$C_l = \frac{F_L}{0.5\rho U_\infty^2 D}, \tag{6.6}$$

$$C_d = \frac{F_D}{0.5\rho U_\infty^2 D}. \tag{6.7}$$

The initial density is taken as $\rho = 1.0$. At $Re = 20$ and $Re = 40$ the freestream velocity is set to $U_\infty = 0.1$ and to $U_\infty = 0.04$ at $Re = 100$ and $Re = 150$ respectively. The computational domain is $50D \times 50D$ with nine levels of grid refinement. The inlet, outlet and free slip boundary conditions, as described in Chapter 2, have been used for the domain boundaries. The region around the cylinder is $4D \times 2D$ with a uniform mesh of 405×205 grid points.

6.2.1 Steady flow over a circular cylinder

For $Re = 20$ and 40, a development of two symmetric, stationary recirculating eddies is observed behind the cylinder. The wake length L_w or, recirculation length, is defined as $L_w = 2L/D$ where, L is the distance from the rearmost point of the cylinder to the end of the wake, as shown is Fig. 6.3.

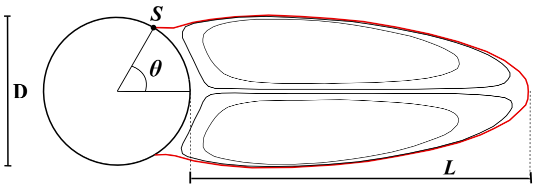
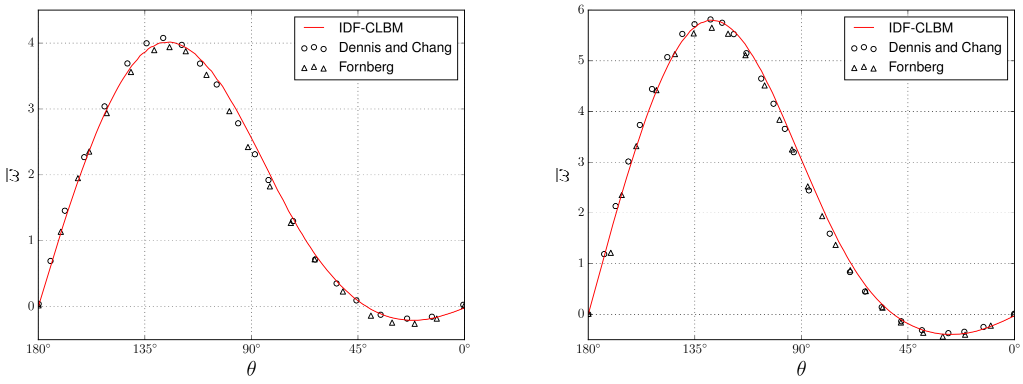


Figure 6.3: Wake length L and separation angle θ_s for steady flow over a circular cylinder

The separation angle θ_s is defined as the angle between the rearmost point of the cylinder and the point s on the cylinder surface where the shear stress is zero. The drag coefficient C_d, the wake length L_w and the separation angle θ_s are compared with other numerical and experimental results [134, 135, 136, 104, 137] in Table

Table 6.1: Comparison of Drag coefficient, wake length L_w and separation angle θ_s for steady flow over a circular cylinder at $Re = 20$ and 40.

Case	References	C_d	L_w	θ_r (deg)
$Re = 20$	[134] - Finite difference, NSE	2.045	1.88	43.7
	[135] - Body-fitted grid, NSE	2.000	1.82	-
	[136] - Interpolation-supplemented, LBM	2.152	1.842	42.96
	[104] - Implicit velocity correction IBM	2.091	1.86	-
	[137] - Vorticity streamfunction, NSE	2.053	1.786	43.37
	Present - IDF-CLBM	2.019	1.857	43.06
	Present - MDF-CLBM	2.005	1.882	43.45
$Re = 40$	[134] - Finite difference, NSE	1.522	4.69	53.8
	[135] - Body-fitted grid, NSE	1.498	4.48	-
	[136] - Interpolation-supplemented, LBM	1.499	4.49	52.84
	[104] - Implicit velocity correction, IBM	1.565	4.62	-
	[137] - Vorticity streamfunction, NSE	1.550	4.357	53.34
	Present - IDF-CLBM	1.524	4.61	53.12
	Present - MDF-CLBM	1.522	4.567	53.41



(a) **(b)**

Figure 6.4: Physical vorticity distribution on the surface of the cylinder at (a): $Re = 20$ and (b): $Re = 40$.

6.1. Both immersed boundary treatments agree well with the other results in the literature.

The physical vorticity $\overline{\omega}$, around the surface of the cylinder at $Re = 20$ and $Re = 40$ is compared with the results of Fornberg [135] and Dennis [134] in Fig. 6.4. All dimensionless quantities in the LBM can be converted into physical quantites as $\overline{\chi} = \chi C_\chi$,where C_χ is a conversion factor and χ is the dimensionless quantity. Therefore, $\overline{\omega}$ can be computed as

$$\overline{\omega} = \omega C_\omega = \frac{\omega}{C_t} = \frac{\omega \overline{u}}{u C_L^2}, \tag{6.8}$$

where C_ω, C_t and C_L are the conversion factors for the vorticity, the time and the characteristic length of the physical problem respectively. First, $\overline{\omega}$ is computed in the 2D computational domain using $\overline{\omega} = \left(\frac{\partial \overline{u}_y}{\partial x} - \frac{\partial \overline{u}_x}{\partial y}\right)$, where $(\overline{u}_x, \overline{u}_y)$ is the physical velocity, and then interpolated on the immersed surface.

6.2.2 Computational efficiency and iteration convergence rate

The relative computational effort of the bulk flow (CLBM), the immersed boundary method and the main steps in the IBM are presented in Table 6.2, based on two runs on each of three different architectures. In all cases the relative efforts are the same. The computational grid consists of 10^6 points and the immersed body is discretised using 312 points. $N = 20$ iterations are used in both immersed boundary methods. The two additional steps in the IDF scheme increase its computational cost over the MDF scheme by 32.8%. However, since both steps have an influence range close to the boundary and are not related to the number of IB points, the computational cost would not increase if a finer representation of the boundary was selected. The overall computational overhead of the IDF-CLBM over the MDF-CLBM is limited to 2.1%. As a final remark, increasing the number of iterations in the IFD scheme from $N = 1$ to $N = 20$ leads to a 6.4% increase in the overall computational cost. However, as shown

Table 6.2: Computational time percentages of the bulk flow (CLBM), the immersed boundary method and the main IBM steps shown in Fig. 5

	CLBM (%)	IBM (%)	VI (%)	FS (%)	DF (%)	Streaming (%)
IDF-CLBM	93.6	6.40	0.95	2.46	1.17	0.47
MDF-CLBM	95.4	4.61	0.93	2.41	-	-

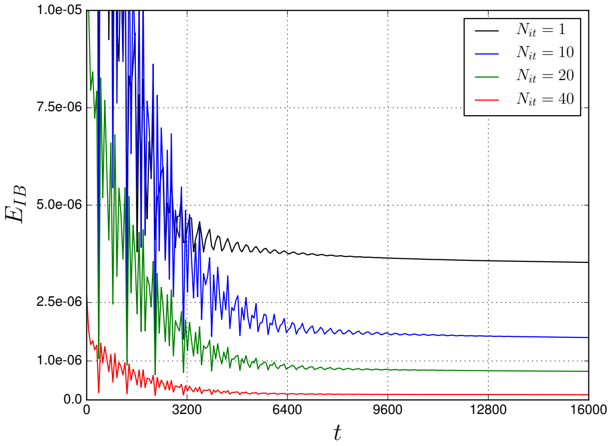


Figure 6.5: Iteration convergence rate of the average boundary error E_{IB} at different numbers of iterations for the flow around the circular cylinder at $Re = 40$. The average boundary error E_{IB} is computed using Eq. (6.3), where the summation is taken over all Lagrangian points.

in Fig. 6.5, that increase in N significantly reduces the boundary error by an order of magnitude. It should be noted that the errors shown in Fig. 6.5 are case specific and are not representative of all the cases presented in this study.

A more detailed breakdown of the computational costs of the present algorithm is presented in Appendix A.3.

Table 6.3: Comparison of lift coefficient, drag coefficients and Stouhal number for unsteady flow over a circular cylinder at $Re = 100$.

Case	References	$Avg.C_d$	C_l	St
$Re = 100$	[138] - Experiment	-	-	0.166
	[69] - 2^{nd} order IBM-NSE	1.447	± 0.330	0.165
	[139] - Body-fitted method, NSE	1.35	± 0.339	0.164
	[104] - Implicit velocity correction IBM	1.364	± 0.344	0.163
	[76] - Higher order IBM-NSE	1.34	± 0.315	0.164
	[79] - Exterior sharp direct forcing LBM	1.336	± 0.329	0.165
	Present IDF-CLBM	1.334	± 0.331	0.165
	Present MDF-CLBM	1.330	± 0.327	0.164

6.2.3 Unsteady flow over a circular cylinder

For $Re = 100$ and 150, vortices are shed from the body. The Strouhal number is defined as

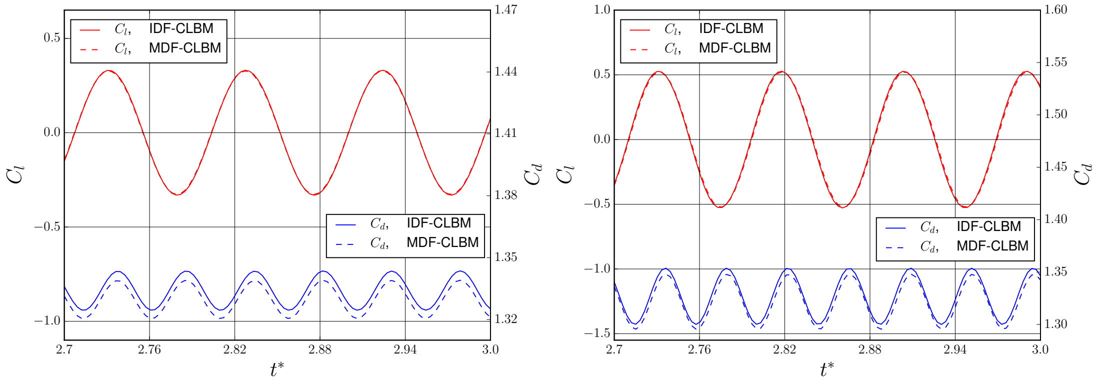
$$St = \frac{f_d D}{U_\infty}, \tag{6.9}$$

where f_d is the shedding frequency. In order to trigger the vortex shedding, the cylinder is placed slightly asymmetrically at $(x_c, y_c) = (L_x/2, (L_y - \delta x)/2)$, where δx is the fine grid spacing and (L_x, L_y) is the domain size in fine grid lattice units. Tables 6.3 and 6.4 present average drag coefficients, minimum and maximum values of lift coefficients and Strouhal numbers. The results are compared with other numerical schemes and experiments[138, 69, 139, 104, 76, 79]. Both methods are in good agreement with the literature. Fig. 6.6 shows the time evolution of the lift and drag coefficients for the IDF-CLBM and the MDF-CLBM. The MDF-CMLB computed a slightly lower drag coefficient than the IDF-CLBM. The relative time t^* is defined as $t^* = t^n U_\infty / D$, where t^n is the current timestep.

At $Re = 150$, the IDF-CLBM drag coefficient has a better agreement with the

Table 6.4: Comparison of lift coefficient, drag coefficients and Stouhal number for unsteady flow over a circular cylinder at $Re = 150$.

Case	References	$Avg.C_d$	C_l	St
$Re = 150$	[138] - Experiment	-	-	0.183
	[69] - 2^{nd} order IBM-NSE	1.44	-	0.184
	[139] - Body-fitted method, NSE	1.334	± 0.530	0.182
	[79] - Exterior sharp direct forcing, LBM	1.312	± 0.513	0.184
	Present IDF-CLBM	1.327	± 0.528	0.182
	Present MDF-CLBM	1.322	± 0.522	0.183



(a) **(b)**

Figure 6.6: Time evolution of lift and drag coefficients for the flow around a circular cylinder at (a): $Re = 100$ and (b): $Re = 150$.

body-fitted method of Liu *et al.* [139] with a relative error $< 1\%$. Overall, the no-slip boundary condition on the immersed surface is well satisfied and a momentum exchange between the interior and the exterior to the boundary fluid domains, that would increase the computed force is minimised.

The time averaged physical vorticity $\overline{\omega^*}$, around the surface of the cylinder at $Re = 100$ is compared with the results of Fornberg [135] and Dennis and Chang [134] in Fig. 6.7. The present IDF-CLMB agrees well with the body-fitted method of Fornberg [135] and the finite difference solver of Dennis and Chang [134].

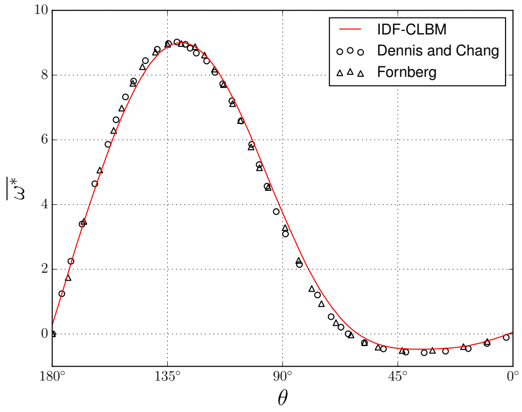


Figure 6.7: Time averaged physical vorticity distribution on the surface of the cylinder at $Re = 100$

6.3 Flow over a NACA-0012 airfoil

The flow around a NACA-0012 airfoil is investigated in this section. Because of the streamlined shape of the airfoil, a body-fitted method would be a usual approach in order to get high accuracy in the aerodynamic coefficients. This issue is addressed by further refining the region around the airfoil, while maintaining a uniform Cartesian grid. The computational domain is $100c \times 100c$ [140], where $c = 400\delta x$ is the chord of the airfoil and $\delta x = 1.0$ is the uniform grid spacing. Ten levels of grid refinement are used in order to ensure that the boundaries will not deteriorate the accuracy of the solution. The inlet, outlet and free slip boundary conditions, as described in Chapter 2, have been used for the domain boundaries. The region around the airfoil is $1.5c \times 0.5c$ with a uniform mesh of 605×205 grid points.

Four different test cases are studied for the flow around the NACA-0012 airfoil as shown in table 6.5. Similar to the flow around the circular cylinder, the density is initialised with $\rho = 1.0$. In table 6.5, N_{iter} is the number of iteration in the IDF scheme.

6.3.1 Steady flow over a NACA-0012 airfoil

The first test case studies the steady flow around the NACA-0012 airfoil at $Re = 500$ at $AoA = 0^o$. The time averaged pressure coefficient can be computed as

$$C_p = \frac{p(\mathbf{X}_l) - p_\infty}{0.5\rho U_\infty^2}, \tag{6.10}$$

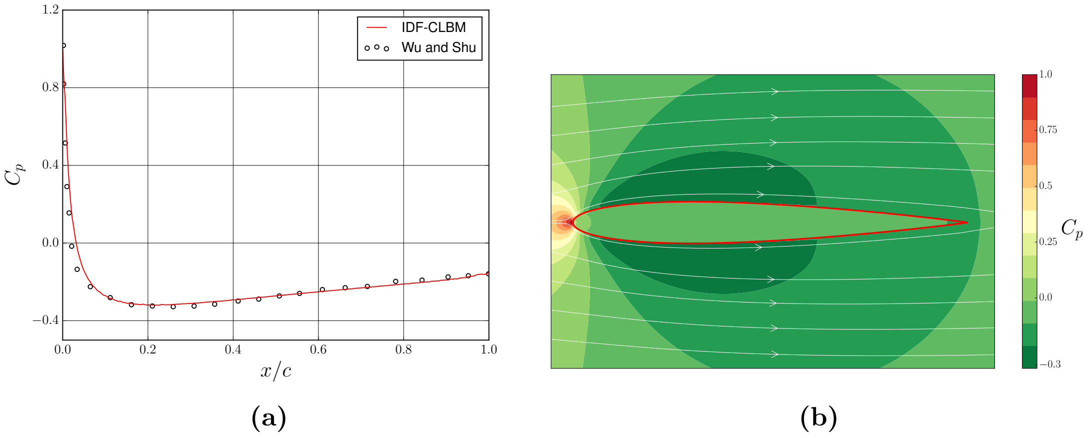
where p_∞ is the freestream pressure. The pressure $p(\mathbf{X}_l)$ on the surface of the airfoil can be interpolated from the surrounding fluid nodes. In the present study, only the exterior to the boundary fluid nodes are used in the interpolation. The distribution of the pressure coefficient along the surface of the airfoil and the pressure contours along with the streamlines are presented in Fig. 6.8. The results are in very good agreement with the implicit velocity correction-based method of Wu and Shu [104].

Fig. 6.9 indicates the boundary layer velocity profile at different positions with respect to the chord of the airfoil ($x/c = 0.0, 0.25, 0.5, 0.75, 1.0$). The time averaged velocity profile over 20000 time-steps at a steady state flow is used. The results are in good proximity with the ones reported by Imamura *et al.* [2] using a LBM on a generalized coordinate system.

The boundary layer thickness δ, as well as the displacement thickness δ_1 and the momentum thickness δ_2, are also computed and compared with the Blasius solution for a flat plate at the same Re, as shown in Fig. 6.11. The boundary

Table 6.5: Test cases for the NACA-0012 airfoil

Case	Re	AoA (deg)	c (δx)	U_∞	Wake	N_{iter}
1	500	0	400	0.1	Steady	20
2	1000	10	400	0.04	Unsteady	20
3	5000	10	400	0.02	Unsteady	40
4	100000	10	800	0.01	Unsteady	80



(a) (b)

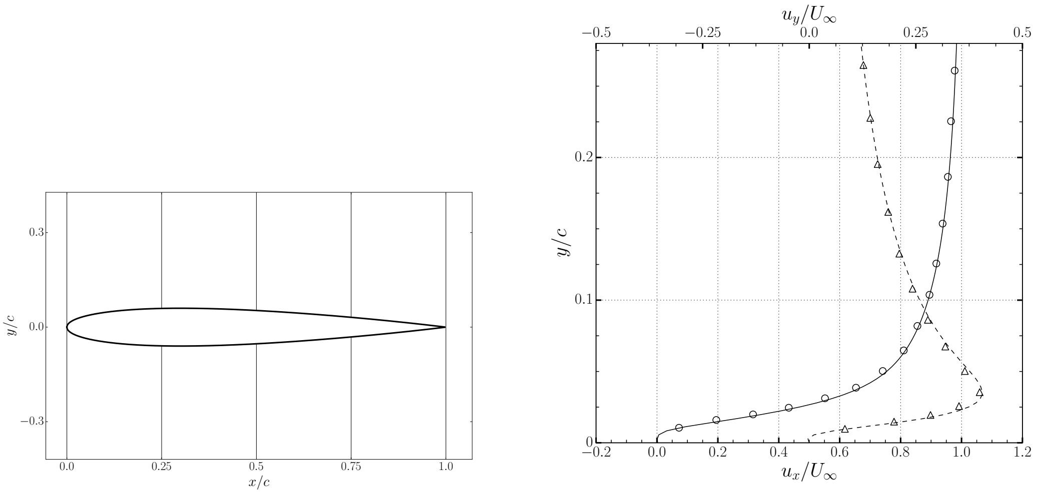
Figure 6.8: (a): Comparison of the pressure coefficient C_p distribution along the NACA-0012 airfoil at $Re = 500$ and $AoA = 0^o$. (b): Streamlines and pressure coefficient contour for the flow around the NACA-0012 airfoil at $Re = 500$ and $AoA = 0^o$.

layer thickness δ is computed using the vorticity decay criterion. As shown in Fig. 6.12, the vorticity ω outside the edge of the boundary layer (red line) is negligible and the flow is almost irrotational with $\nabla \times \mathbf{u} = 0$. The computational procedure used in the present study is as follows. First, the vorticity is interpolated on the wall normal directions $\mathbf{n}_l$ for each Lagrangian marker l on the immersed body. The maximum value of the vorticity $\|\omega\|_{\mathbf{n}_l}$ for each $\mathbf{n}_l$ is stored. Finally, δ is defined as the distance from the wall where $\omega_{\mathbf{n}_l}$ has decayed to a small fraction of the maximum vorticity $\|\omega\|_{\mathbf{n}_l}$ near the wall as

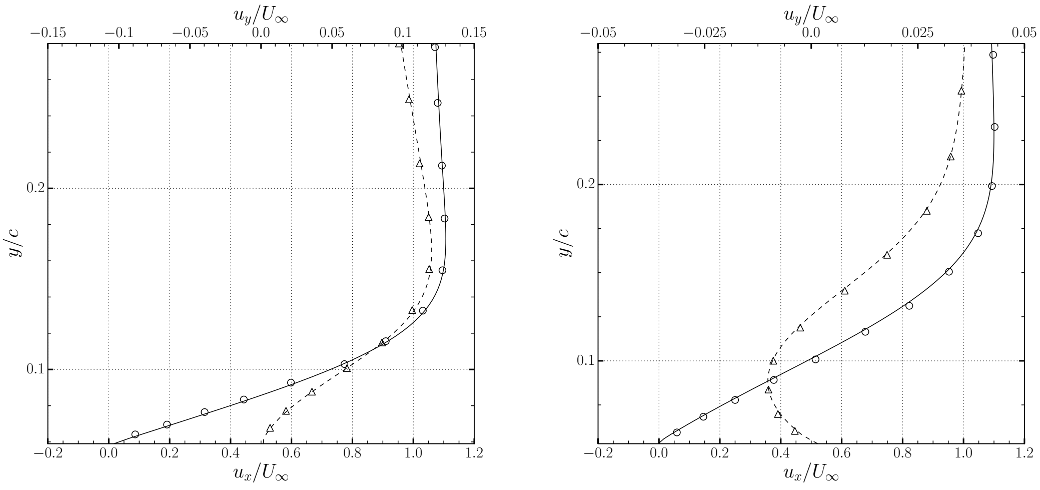
$$\hat{\omega}_{\mathbf{n}_l} = \frac{\omega_{\mathbf{n}_l}}{\|\omega\|_{\mathbf{n}_l}} \leq \epsilon, \tag{6.11}$$

where $\epsilon = 0.02$ is chosen based on the flat plate correlation. Knowing δ, the displacement thickness δ_1 and the momentum thickness δ_2 are computed as

$$\delta_1 = \int\limits_0^\delta \left(1 - \frac{u_{\mathbf{n}_l}}{U_{\mathbf{n}_l,\delta}}\right) dy, \qquad \delta_2 = \int\limits_0^\delta \frac{u_{\mathbf{n}_l}}{U_{\mathbf{n}_l,\delta}} \left(1 - \frac{u_{\mathbf{n}_l}}{U_{\mathbf{n}_l,\delta}}\right) dy, \tag{6.12}$$



(a) Cross sections **(b) x/c = 0.00**

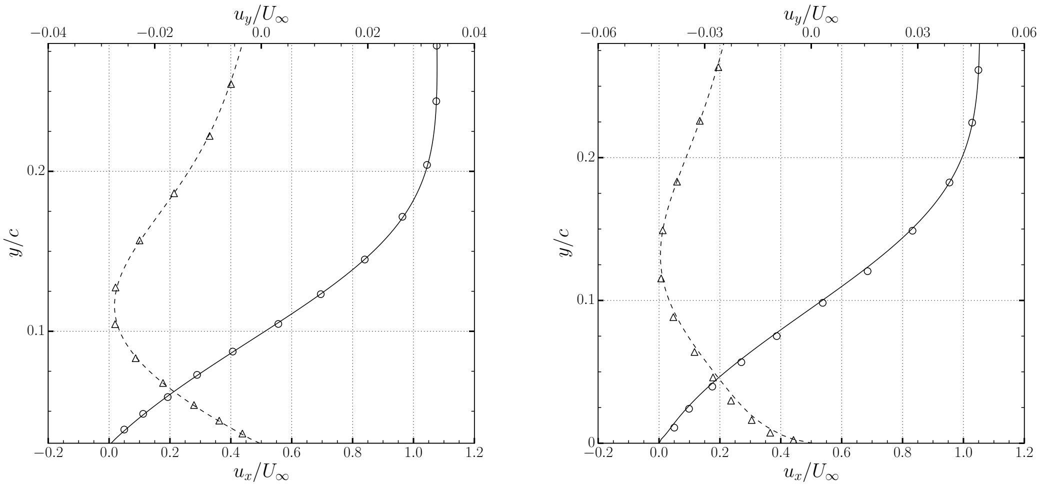


(c) x/c = 0.25 **(d) x/c = 0.50**

Figure 6.9: Comparison of the time averaged velocity profile of NACA 0012 at various cross sections x/c. The normal and dashed lines are the u_x/U_∞ and u_y/U_∞ respectively, of the IDF-CLBM. The results from Imamura [2] are plotted with circles and triangles.

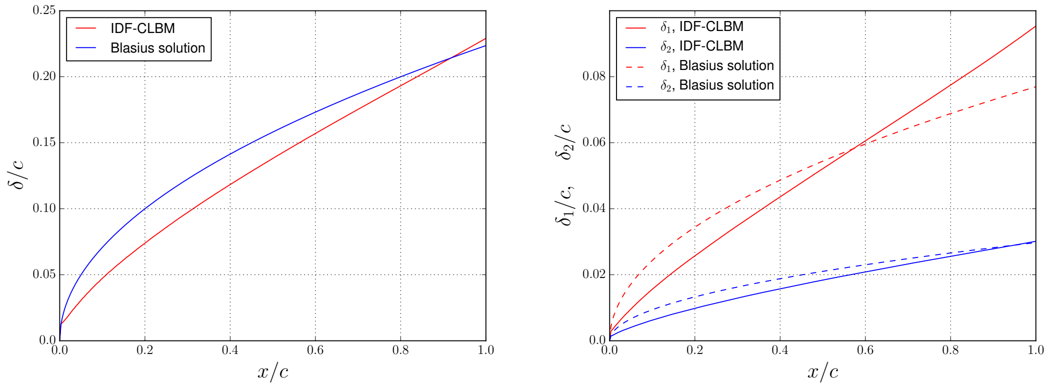
where $u_{\mathbf{n}_l}$ and $U_{\mathbf{n}_l,\delta}$ are the tangential components of the computed velocity and the outer velocity on the normal to the wall directions $\mathbf{n}_l$.

Fig. 6.11a shows that the accelerated flow from the leading edge leads in a



(a) x/c = 0.75 **(b) x/c = 1.00**

Figure 6.10: Fig. 6.9 continued.



(a) **(b)**

Figure 6.11: (a): Boundary layer thickness with Blasius solution (b): Discplacement (δ_1) and momentum (δ_2) thickness with Blasius solution.

thinner boundary layer than the flat plate. However, at about 90% of the chord, the boundary layer has fully recovered its thickness due to the adverse pressure gradient. In Fig. 6.11b, a similar behaviour is observed for the momentum thickness, δ_2. The displacement thickness recovers at approximately 55% of the chord length. Fig. 6.11 along with Fig. 6.9 demonstrate the

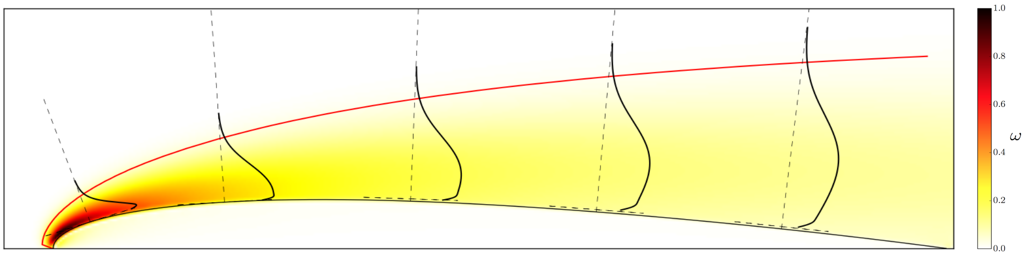


Figure 6.12: Normalised absolute vorticity field around the NACA-0012 at $Re = 500$ and boundary layer thickness δ (red line). The black lines show the vorticity $\omega_{\mathbf{n}_l}$, scaled with the maximum value $\|\omega\|_{\mathbf{n}_l}$, on the normal directions $\mathbf{n}_l$.

accuracy of the present IBM in computing the boundary layer profile around an airfoil at low Reynolds number flows. Accurately computing δ, δ_1 and δ_2 is essential in aerodynamic applications in understanding frictional dissipation effects, momentum deficits in the boundary layer and optimizing the aerodynamic shape of the foil.

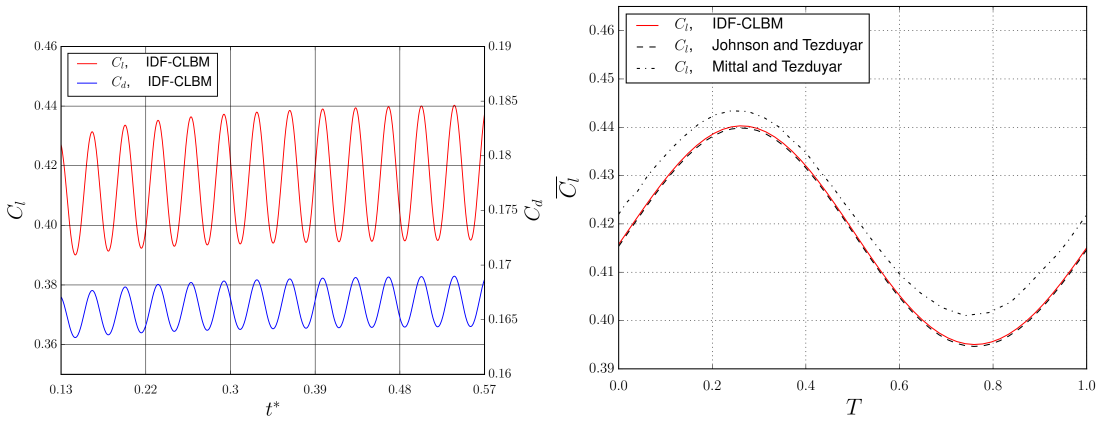
6.3.2 Unsteady flow over a NACA-0012 airfoil

At $Re = 1000$ and $AoA = 10^o$, the computed Strouhal number is 0.861, which compares well with the value of 0.862 reported by Mittal and Tezduyar [140] and the value of 0.86 reported by Johnson and Tezduyar [141]. Fig. 6.13a, shows the time evolution of the lift and drag coefficients. The time averaged, over the last four periods, lift coefficient is compared with the reported values in [140] and [141] as shown in Fig. 6.13b. Our results agree well with the ones reported in [141]. Mittal and Tezduyar [140] reported a 2% higher value of the lift coefficient.

Fig. 6.14 shows the boundary layer thickness around the NACA-0012 airfoil. The time averaged vorticity over ten oscillation periods is used to illustrate the average behaviour of the unsteady flow around the NACA0012 airfoil at Re=1000. Effectively, the vorticity distribution along the normal directions along with the skin friction values on the surface may be used to identify properties of the boundary layer evolution around the airfoil and lead to deeper understanding of its behaviour at unsteady flow cases.

At $Re = 5000$ and $AoA = 10^o$, the presence of stronger vortices on the surface of the airfoil result in higher average values of the aerodynamic coefficients. Fig. 6.15a shows the time evolution of the lift and drag coefficients. In order to accurately capture the flow characteristics, a lattice freestream velocity $U_\infty = 0.02$ is used. The magnitudes of the temporarily fluctuating components of the aerodynamic coefficients are also higher for $Re = 5000$. The lift coefficient computed with the IDF-CLBM agrees well with the one reported in [140], as show in Fig. 6.15b. At both $Re = 1000$ and $Re = 5000$, the magnitude of oscillation of the lift coefficient reported in [140] is slightly lower than the one computed with IDF-CLBM.

Fig. 6.17 shows a sequence of frames for the instantaneous vorticity around the



(a) (b)

Figure 6.13: (a): Time evolution of lift and drag coefficients for the flow around a NACA-0012 airfoil at $Re = 1000$ and $AoA = 10^o$, (b): Averaged lift coefficient $\overline{C_l}$, over four oscillations with period T.

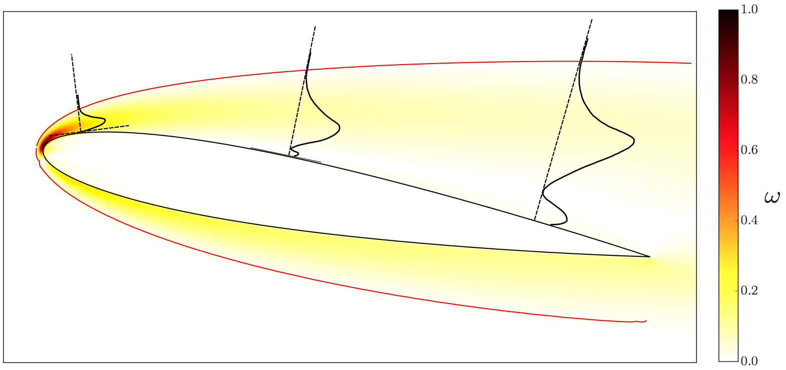
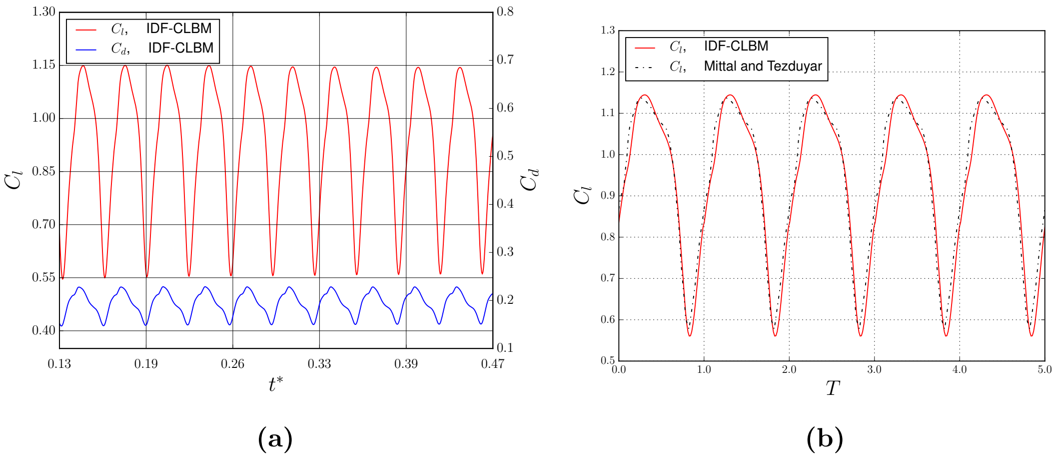


Figure 6.14: Normalised absolute vorticity field around the NACA-0012 at $Re = 1000$ and boundary layer thickness δ (red line). The black lines show the vorticity $\omega_{\mathbf{n}_l}$, scaled with the maximum value $||\omega||_{\mathbf{n}_l}$, on the normal directions $\mathbf{n}_l$.

airfoi during one period of the lift coefficient. The first frame (6.16a) corresponds to the minimum lift coefficient, whereas the fourth frame (6.16d) corresponds to the maximum lift coefficient. It is noticed that the interaction between the vortices shed from the upper and the lower surfaces of the airfoil generates an upwards moving wake with counter-rotating vortices. A similar conclusion is reported in [140]. The computed Strouhal number, based on the dominant frequency is 0.681 which agrees well with the value of 0.685 of Mittal and Tezduyar [140].

At $Re = 10^5$ and $AoA = 10^o$, a direct comparison with other numerical simulations is not feasible. The freestream lattice velocity is set to $U_\infty = 0.01$ and the chord of the airfoil at $c = 800\delta x$ with $\delta x = 1.0$. Fig. 6.16 shows the time histories of the aerodynamic coefficients acting n the airfoil.

Table 6.6 summarises some statistical characteristics of the time evolution of the lift and drag coefficients. The results are compared with the ones reported in Mittal and Tezduyar [140] and good agreement is observed on the drag coefficient. However, differences are observed in the lift coefficient. Although the mean and maximum values are closely related, a divergence is observed in the standard



(a) (b)

Figure 6.15: (a): Time evolution of lift and drag coefficients for the flow around a NACA-0012 airfoil at $Re = 5000$ and $AoA = 10^o$, (b): Lift coefficient over five oscillation periods.

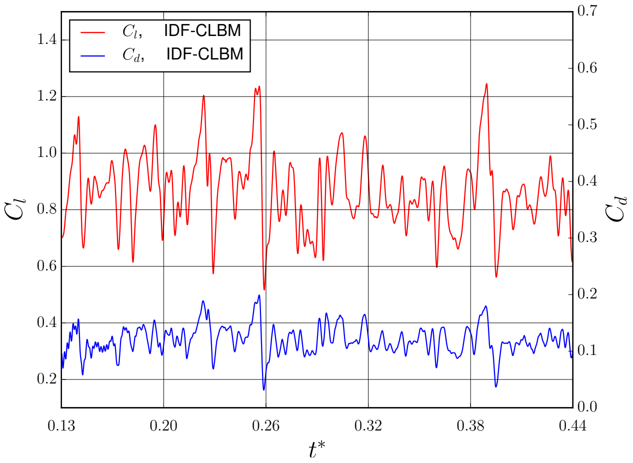


Figure 6.16: Time evolution of lift and drag coefficients for the flow around a NACA-0012 airfoil at $Re = 10^5$ and $AoA = 10^o$.

deviation and the minimum value. The quantile-quantile (Q-Q) plots in Fig. 6.18 compare the data of the lift and drag coefficients obtained with the IDF-CLBM with the data reported in Mittal and Tezduyar [140]. Good statistical proximity is observed for the drag coefficient with a 5% average divergence between the two

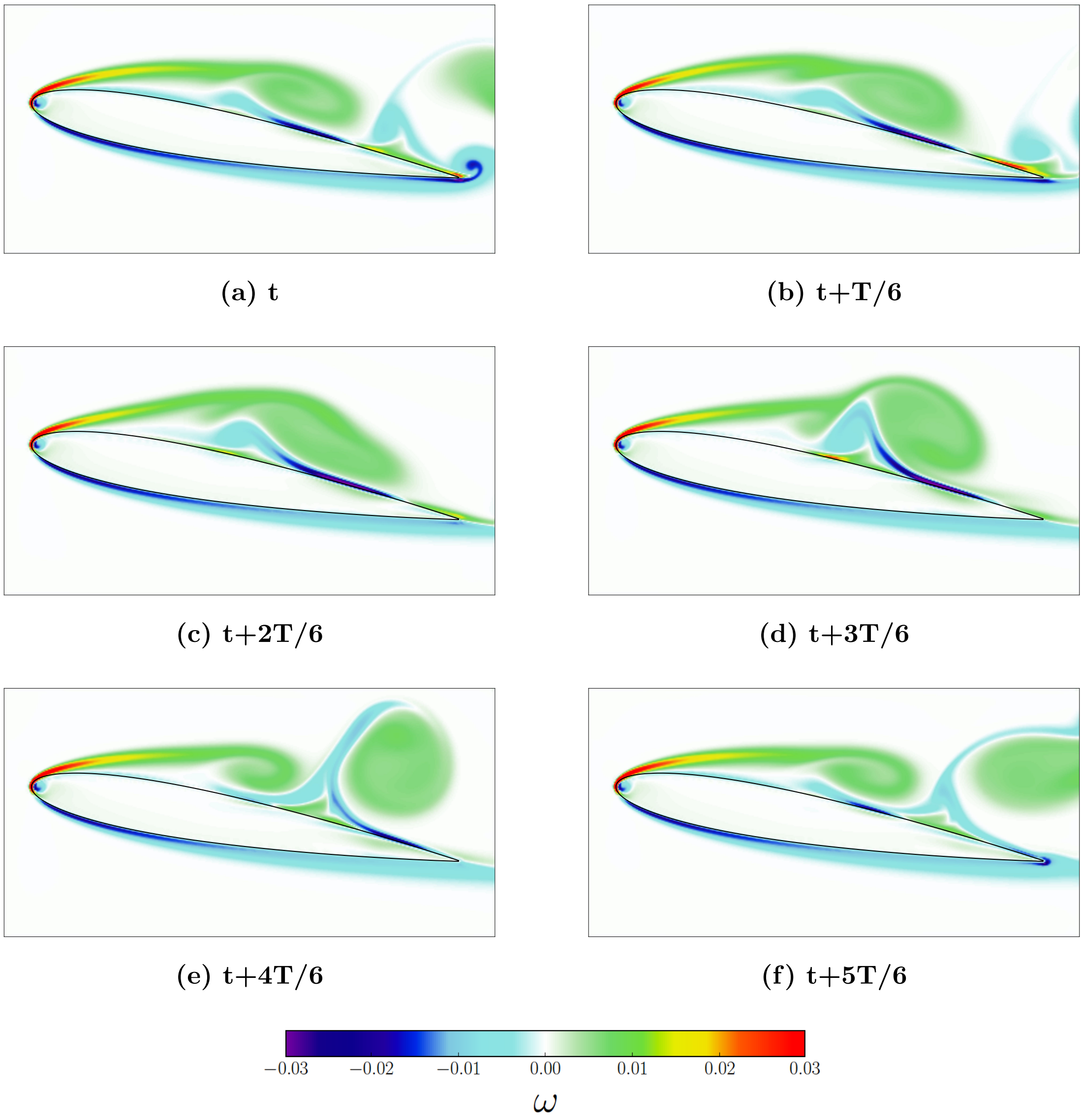
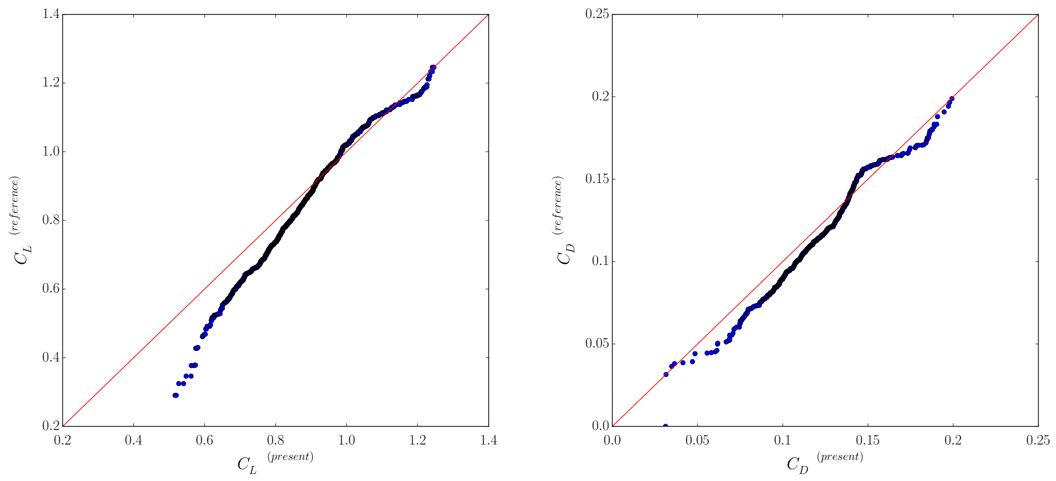


Figure 6.17: Instantaneous vorticity around the NACA-0012 airfoil at $Re = 5000$ and $AoA = 10^o$ for a period of oscillation of the lift coefficient.

data sets. However, as seen in Fig. 6.18a, below the mean value, the data range for the lift coefficient significantly diverges.

According to Mittal and Tezduyar [140], flows at such high Reynolds numbers are considered turbulent in nature. It should be noted that Case 4 could be



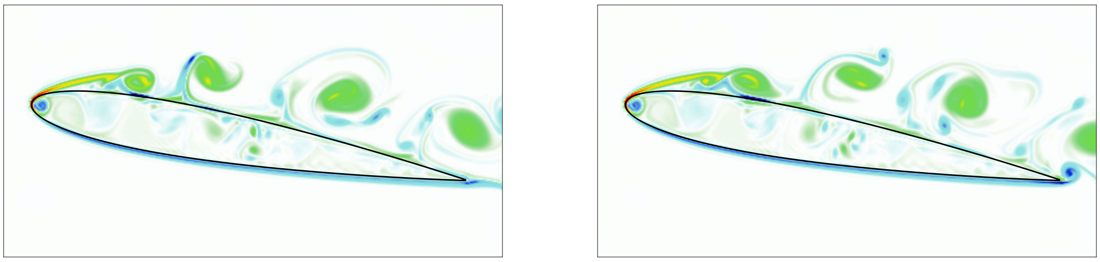
(a) **(b)**

Figure 6.18: (a): Q-Q plot for the lift coefficient, (b): Q-Q plot for the drag coefficient.

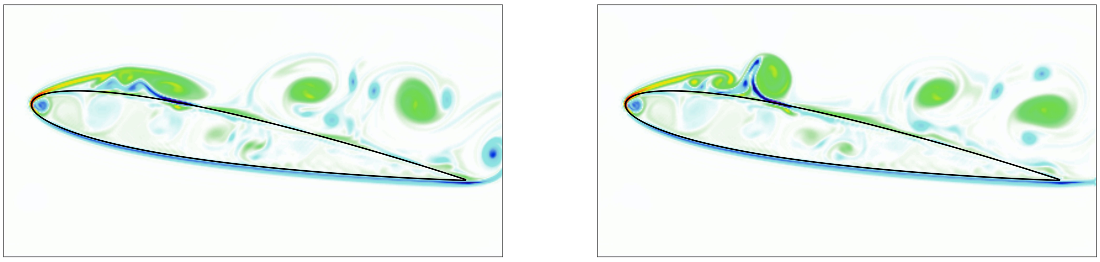
Table 6.6: Statistics for the time evolution of the lift and drag coefficients for the flow around a NACA-0012 airfoil at $Re = 10^5$ and $AoA = 10^o$.

	Reference	Mean	St. deviation	Min	Max
C_L	Mittal and Tezduyar [140]	0.82705	0.15917	0.30035	1.24603
	Present IDF-CLBM	0.85577	0.12797	0.51671	1.24583
C_D	Mittal and Tezduyar [140]	0.11457	0.02995	0.03149	0.19888
	Present IDF-CLBM	0.11724	0.02776	0.03139	0.20724

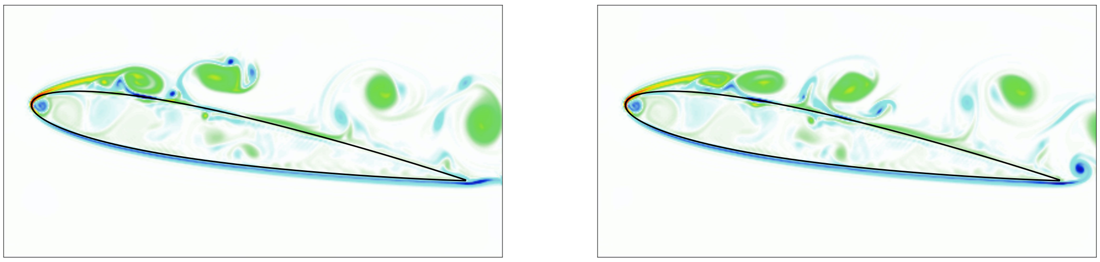
considered as a limit to the current IDF immersed boundary algorithm. Although the no-slip boundary condition is still satisfied, the effect of the force spreading operation in the immersed boundary scheme and the first order of accuracy of the Dirac delta function distorts the accuracy of the solution around the immersed boundary. This leads to discontinuities of the velocity gradient on the boundary and decreases the order of accuracy of the solution. This can also be verified by the over-predicted values of the lift coefficient in Fig. 6.18a. The issues will be carefully addressed in Chapter 8, where a different strategy for removing the



(a) t=0.433 **(b) t=0.436**



(c) t=0.438 **(d) t=0.440**



(e) t=0.442 **(f) t=0.444**

Figure 6.19: Instantaneous vorticity around the NACA-0012 airfoil at $Re = 10^5$ and $AoA = 10^o$.

discontinuities is proposed. As a final remark, the instantaneous vorticity at $t^* = 0.433, 0.436, 0.438, 0.44, 0.442, 0.444$ is shown in Fig. 6.19. It can be observed that separation of the flow on the upper surface of the airfoil occurs very close to the leading edge.

Case 4, demonstrates the robustness of our numerical scheme at high Reynolds number flows. In contrast to the previous cases, an unsteady motion of the internal fluid is observed at $Re = 10^5$, as shown in Fig. 6.19. The time derivative of the linear momentum of the internal fluid results in an internal force $\mathbf{F}^{int}$ given by

$$\mathbf{F}^{int}(t) = \rho \frac{d}{dt} \int\limits_{\mathbf{x} \in \Omega(t)} \mathbf{u}(\mathbf{x}, t) d\mathbf{x}. \tag{6.13}$$

The effect of the internal forces is more significant in moving boundary applications. In highly unsteady flow cases, compensating the internal forces in the computation of the aerodynamic forces is crucial, as will be shown in Chapter 7.

Chapter 7

Computational Results: Flows around moving and wilfully deformed objects

In this chapter, we extend the application of the IDF-CLBM solver to flows around moving and wilfully deformed bodies with prescribed motions. Emphasis is given on the influence of the internal mass on the computation of the aerodynamic forces including deforming boundary applications where the rigid body approximation is no longer valid. Both the rigid body and the internal Lagrangian points approximations are examined. The resulting solver has been applied to viscous flows around an in-line oscillating cylinder, a pitching foil, a plunging SD7003 airfoil and a plunging and flapping NACA-0014 airfoil. Good agreement with experimental results and other numerical schemes has been obtained. It is shown that the internal Lagrangian points approximation accurately captures the internal mass effects in linear and angular motions, as well as in deforming motions, at Reynolds numbers up to $4 \cdot 10^4$. In all cases, the aerodynamic loads are significantly affected by the internal fluid forces.

7.1 Oscillating circular cylinder in a stationary fluid

First, the flow around an oscillating circular cylinder in a stationary fluid is considered. Dütch *et al.* [3] studied this problem both experimentally and numerically, and other numerical results have been reported in various studies [90, 142, 143, 76]. The computational domain is $100D \times 100D$, where $D = 100lu$ is the diameter of the cylinder and $lu = 1.0$ is the lattice unit. Periodic boundary conditions have been used for the domain boundaries. The cylinder is placed at the center of the domain and the prescribed oscillating motion is described as

$$X_c(t) = A_m \sin(2\pi f t), \tag{7.1}$$

where A_m denotes the amplitude and f is the characteristic frequency of the oscillating motion, as shown in Fig. 7.1. The non-dimensional angle of the periodic motion is ϕ, where $0 \leq \phi \leq 1$. The governing parameters of the flow are the Reynolds number $Re = U_{max}D/\nu$ and the Keulegan-Carpenter number $KC = U_{max}/fD$. The amplitude of the velocity, U_{max}, is derived from Eq. (7.1) as $U_{max} = 2\pi f A_m$. Thus the Keulegan-Carpenter number can be expressed as $KC = 2\pi A_m/D$. We investigate the flow at $Re = 100$ and $KC = 5$. $N_{iter} = 20$ iterations are used in the IBM algorithm.

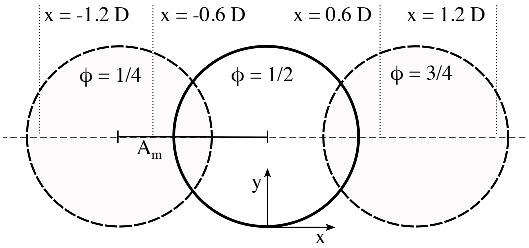


Figure 7.1: Schematic view of the cylinder and the oscillating motion.

The velocity profile at four cross sections ($x = x_1 - 0.6D, x_1 + 0.0D, x_1 + 0.6D, x_1 +$

$1.2D$) with constant $x_1 = L_x/2$ values, where L_x is the size of the domain in the x direction, and three phase angles ($\phi = 180^o, 210^o, 330^o$) are shown in Fig. 7.2.

The experimental results of Dütch *et al.* [3] are used for comparison. Good agreement between the present results and the experimental data is observed. The symmetric and periodic vortex shedding, as well as the two-dimensional behaviour of the flow for the present parameter set up ($Re = 100$ and $KC = 5$) are well captured by the present numerical scheme. Fig. 7.3 shows a histogram of the interpolated velocities inside the cylinder. The mean interpolated velocity is similar to the prescribed velocity. However, as shown in Fig. 7.3b the highest deviations from the prescribed velocity are located near the boundary due to the force spreading kernel in the IBM. In the present IMB-CLBM formulation the motion of the Lagrangian points over Eulerian nodes does not cause any problems or instabilities. However, care must be given on the selection of proper motion characteristics, i.e. period of oscillation in lattice units, and the selection of the interpolation/spreading kernels in the IBM.

The drag coefficient $C_D = 2F_D/\rho U_{max}^2 D$ over one period of oscillation, T, is shown in Fig. 7.4a. The drag force F_D is computed using Eq. (3.27). Both the RBA and the LPA schemes are in good quantitative agreement with the computational results of Dütch *et al.* [3]. However, ignoring the internal mass effects lead to a significant over-prediction of the drag coefficient by approximately 51%, as well as to an important phase difference of 12.1^o. Fig. 7.4b shows the instantaneous vorticity, ω, at four phase angles, $\phi = 150^o, 180^o, 210^o, 330^o$.

Using the semi-empirical Morison equation [144], the time-dependent in-line force F_x can be predicted as

$$F_x(t) = -\frac{1}{2}\rho D C_d \dot{X}_c |\dot{X}_c| - \frac{1}{4}\pi \rho D^2 C_i \ddot{X}_c, \tag{7.2}$$

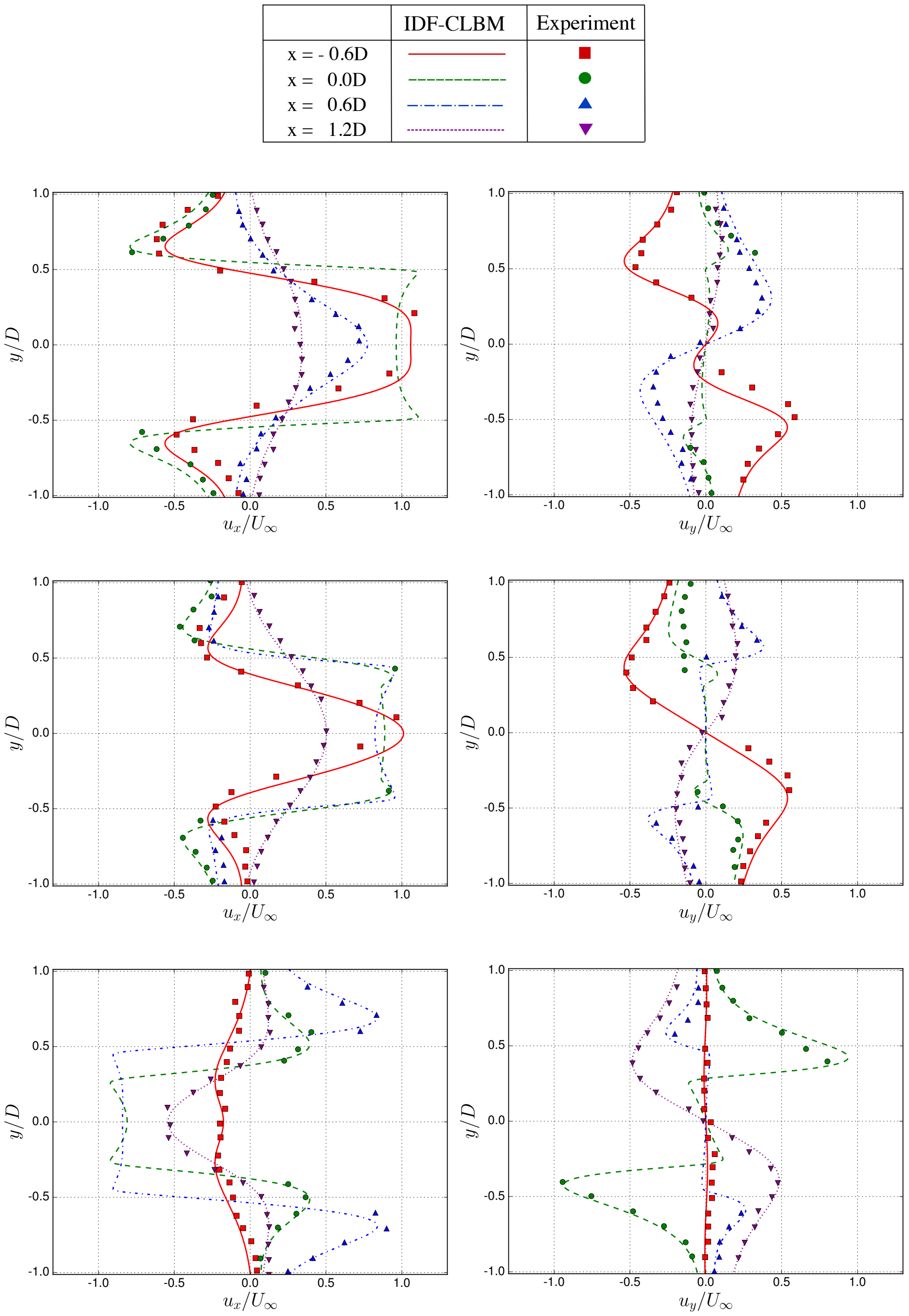
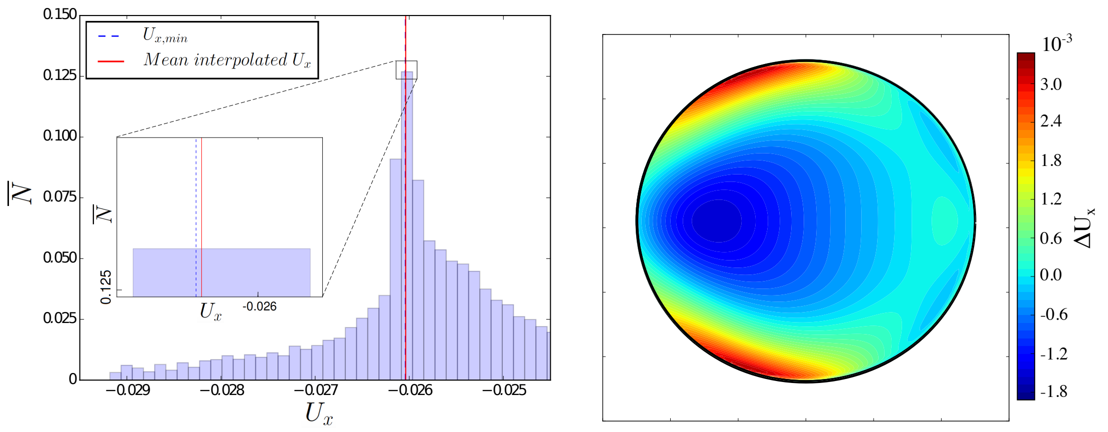
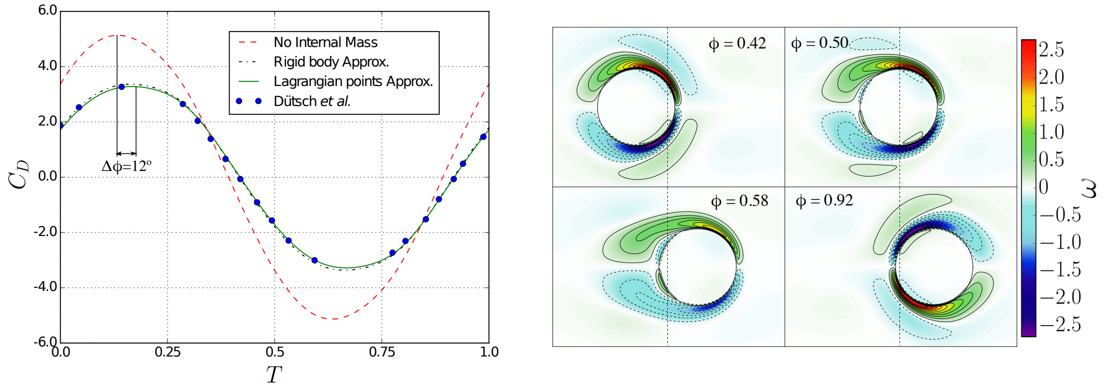


Figure 7.2: Velocity profiles for three phase angles (a) $\phi = 0.5$, (b) $\phi = 0.58$ and (c) $\phi = 0.92$ for $Re = 100$ and $KC = 5$. The experimental results of Dütch *et al.* [3] (symbols) are used for comparison.



(a) **(b)**

Figure 7.3: (a) Histogram of the interpolated velocities inside the cylinder at $\phi = 0^o$ and $x = 0$, (b) Deviation of the interpolated internal velocity from the prescribed velocity at $\phi = 0^o$ and $x = 0$.



(a) **(b)**

Figure 7.4: (a) Drag coefficient over a period of a translationally oscillating circular cylinder for $Re = 100$ and $KC = 5$, (b) Instantaneous vorticity for four phase angles $\phi = 0.42$, $\phi = 0.50$, $\phi = 0.58$ and $\phi = 0.92$.

where C_i is the added-mass coefficient and the dots indicate the time derivatives of Eq. (7.1). Therefore, the motion-averaged drag and added-mass coefficients, C_d and C_i, can be evaluated by a least-square fitting method. The computed values are compared with data from the literature in Table 7.1 and very good agreement is observed.

Table 7.1: Comparison of drag and added-mass coefficients, C_d and C_i, for a translationally oscillating circular cylinder at $Re = 100$ and $KC = 5$.

References	C_d	C_i
Dütch *et al.* [3]	2.09	1.45
Uzunoğlu *et al.* [145]	2.10	1.45
Yuan *et al.* [146]	2.10	1.45
IDF-CLBM	2.09	1.45

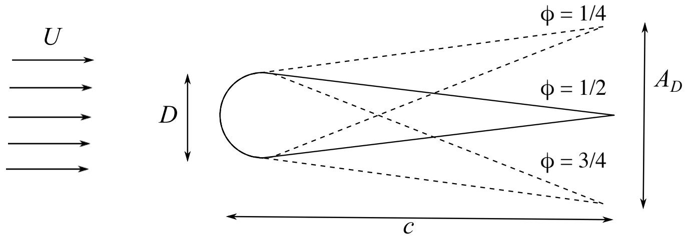
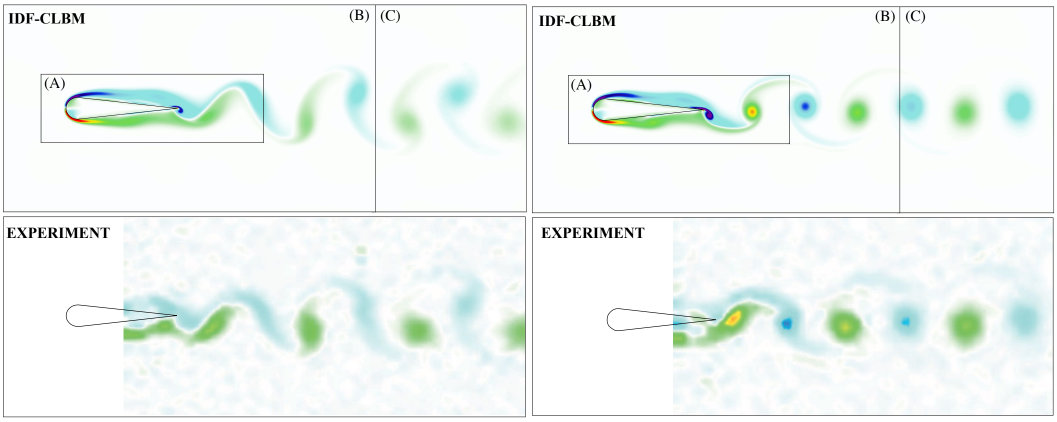


Figure 7.5: Schematic view of the foil and the pitching motion.

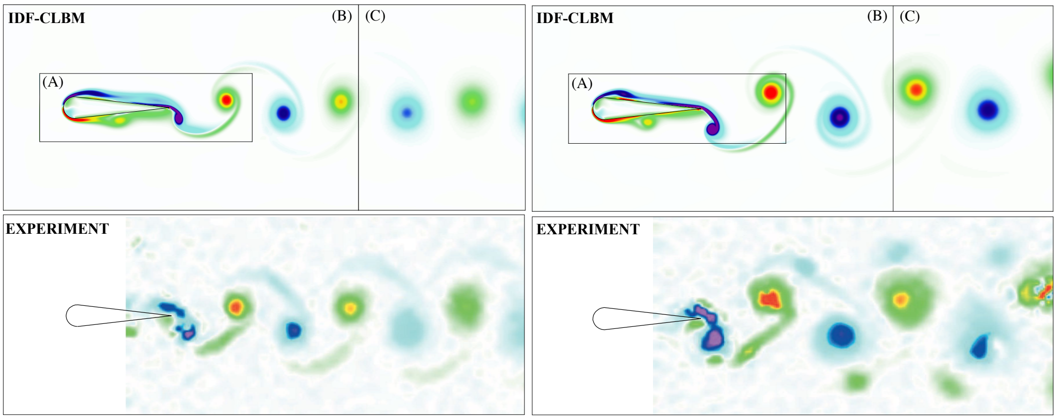
7.2 Drag-thrust transition of a pitching foil

The vortex street of a pitching foil is investigated in this section for a range of flapping frequency-amplitude phases. The governing parameters of the flow are the Reynolds number $Re = U_\infty D/\nu$ and the Strouhal number $Sr = fD/U_\infty$, where $U_\infty = 0.02$ is the freestream velocity, D is the foil thickness and f is the flapping frequency. The thickness to chord ratio is $D/c = 5/23$. According to Diana *et al.* [4, 147], the flapping frequency f is equivalent to the main vortex shedding frequency. The dimensionless flapping amplitude A_D is defined as $A_D = A/D$, where A is the peak-to-peak amplitude, as shown in Fig. 7.5. In the present study, we examine the vortex shedding at $Re_c = 1173$, where $Re_c = U_\infty c/\nu$ is the chord-based Reynolds number and $c = 300lu$ is the chord of the foil. The Strouhal number is set at $S_r = 0.22$ and four dimensionless flapping amplitudes $A_D = 0.36, 0.71, 1.07, 1.77$ are investigated.

Figures 7.6 and 7.7 show instantaneous vorticity fields and average horizontal



(a) $A_D = 0.36$ **(b)** $A_D = 0.71$



(c) $A_D = 1.07$ **(d)** $A_D = 1.77$

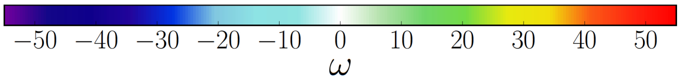
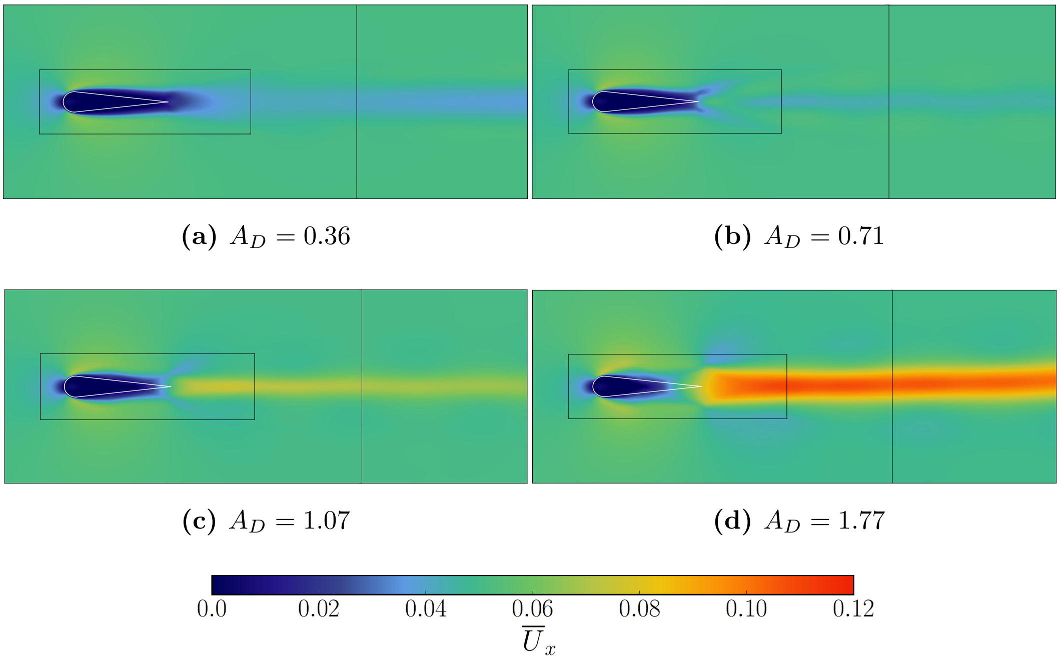


Figure 7.6: Instantaneous vorticity around the pitching airfoil at 0^o angle of attack, $Re = 1173$ and $Sr = 0.22$. (A), (B) and (C) denote the domains with different grid qualities with (A) being the finest domain around the foil.

velocity fields respectively around the foil, for all dimensionless flapping amplitudes. The results are compared with the experimental results of Diana *et al.* [4] and good agreement is observed. The structure of the wake in Fig. 7.6a for a low-amplitude flapping motion, $A_D = 0.36$, resembles the features of a natural Bénard - von Kármán (BvK) vortex street. The velocity deficit behind the foil, as shown in Fig. 7.7a, is a result of the downstream vortices, originating from


(a) $A_D = 0.36$

(b) $A_D = 0.71$

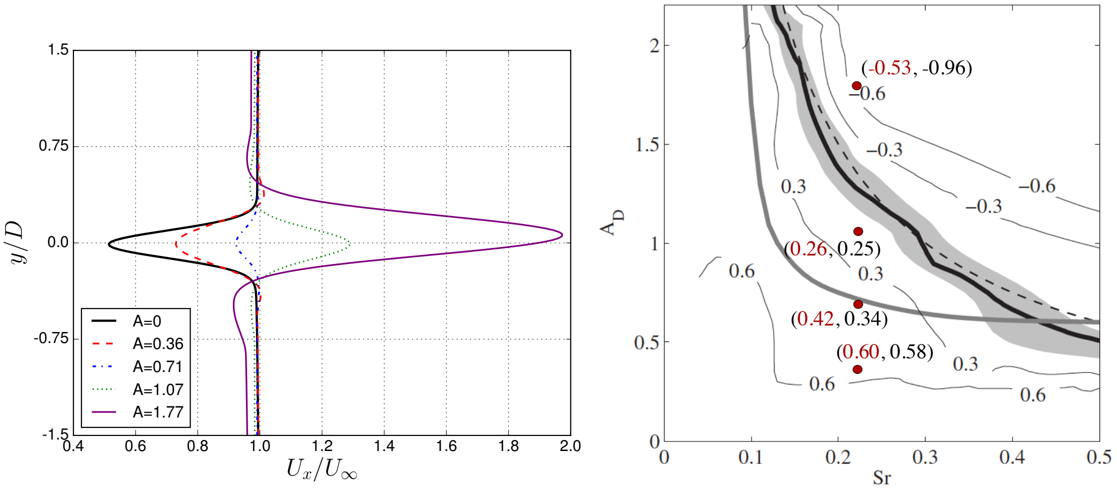
(c) $A_D = 1.07$

(d) $A_D = 1.77$

Figure 7.7: Time averaged horizontal velocity around the pitching airfoil at 0^o angle of attack, $Re = 1173$ and $Sr = 0.22$.

the vorticity on the sides of the foil, that tend to stay on the symmetry line of the wake. Reversal of the vortex position is observed as the dimensionless flapping amplitude increases. For $A_D = 0.71$ the downstream vortices are almost aligned, Fig. 7.6b, decreasing the velocity deficit of the wake significantly, Fig. 7.7b. Further increasing the amplitude $A_D = 1.07, 1.77$ leads to a reverse BvK vortex street, as shown in Figures 7.6c and 7.6d. Vortices created on each side of the foil are organised on the opposite sites of the symmetry line leading to an accelerating flow behind the foil, as shown in Figures 7.7c and 7.7d.

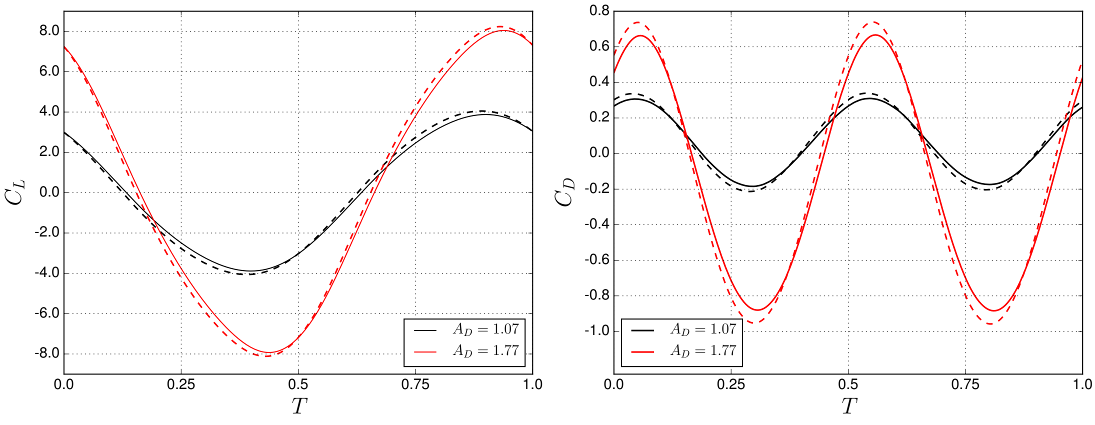
Fig. 7.8a shows the normalised average velocity profile at $X = 12D$ downstream from the trailing edge. Overall, the computed structures of the vortices, as well as the mean velocity fields are in very good agreement with the experimental results of Diana *et al.* [4]. It has been found that at $A_D = 1.77$ the wake is not symmetric leading to net lift force generation, as shown in Fig. 7.9a. For more



(a) **(b)**

Figure 7.8: (a) Normalised average velocity profile at $x = 12D$ downstream from the trailing edge for $A_D = 0, 0.36, 0.71, 1.07, 1.77$, (b) Contours of mean drag coefficient C_D/C_{D0} surface estimated using a momentum balance approach taken from Godoy-Diana *et al.* [4]. The present numerical results of the LPA scheme (red) and the momentum balance approach (black) are included.

information on the symmetry breaking of the reverse BvK vortex street the reader should refer to Diana *et al.* [147]. In Fig. 7.8b, we compare the computed mean drag coefficients with the values obtained by Diana *et al.* [4] using the momentum balance approach in a control volume around the foil. The values obtained with the LPA scheme and with the momentum balance approach are highlighted in red and black respectively. Eq. 3.27 is used for the computation of the mean drag coefficient with the LPA which is then normalised by its value for a rigid foil at zero angle of attack, (C_{D0}). Both methods agree well with the experimental values. However, at $A_D = 1.77$ the drag coefficient value obtained with the momentum balance approach diverges significantly from the experimental value.



(a) **(b)**

Figure 7.9: (a) Lift and (b) drag coefficient over a flapping period for $A_D = 1.07, 1.77$. The solid and dashed lines refer to the computed coefficients with and without the LPA method.

The drag forces for the momentum balance approach are obtained by

$$F_D = \rho U_\infty \int [U_\infty - u(y)] dy, \tag{7.3}$$

where $u(y)$ is the velocity profile at $X = 12D$ downstream from the trailing edge as shown in Fig. 7.8a. Good agreement is observed for $A_D = 0.36, 0.71, 1.07$ where the vortex structure is fully 2-Dimensional. Finally, Fig. 7.9 compares the computed values for the lift and drag coefficients with and without the LPA scheme. Significant differences can be observed both in the magnitude as well as the phase of the oscillating drag coefficient.

7.3 Plunging SD7003 airfoil

The unsteady separated flow encountered by the plunging motion of a SD7003 airfoil under moderate-Reynolds-number conditions ($Re_c = 40000$) is investigated. The airfoil is set at a small static angle of attack $\alpha_0 = 4^o$ in the center of a

$100c \times 100c$ computational domain, where $c = 600lu$ is the chord of the foil. The inlet, outlet and free slip boundary conditions, as described in Chapter 2, have been used for the domain boundaries. The flow over a plunging SD7003 at high-frequency, low-amplitude motions has been recently investigated both numerically and experimentally [6, 148, 149, 150, 5]. The plunging motion is characterised with a reduced frequency $k = \pi f c/U_\infty = 3.93$ and a non-dimensional amplitude $h_0 = \hat{h_0}/c = 0.05$ leading to a maximum excursion of 21.5^o in the induced angle of attack.

A detailed schematic of the motion is presented in Fig. 7.10. This large induced angle of attack leads to unsteady flow fields characterised by leading-edge dynamic stall vortices [6], as shown in Fig. 7.11.

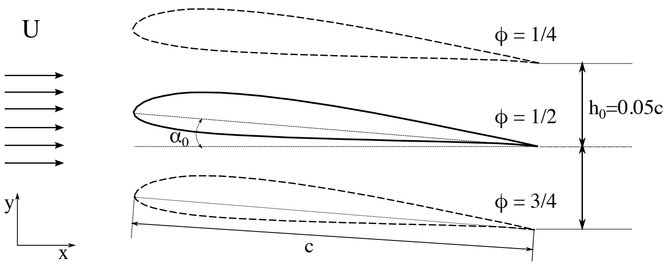


Figure 7.10: Schematic view of the motion of the SD7003 airfoil at a static angle of attack $\alpha_0 = 4^o$.

The motion of the foil is given by

$$y(t) = \hat{h_0} \sin\left[2kF(t)t\right], \tag{7.4}$$

where $F(t) = 1 - e^{at}$ is a ramping function, smoothly transitioning the stationary foil to the plunging motion. The parameter a is set to $a = 9.2$.

Good agreement between the computed and the experimental phase averaged velocity fields in the near wake ($x/c = 1.5$) of McGowan *et al.* [5] is observed in Fig. 7.12. The phase averaged velocity field is computed over 20 cycles. As

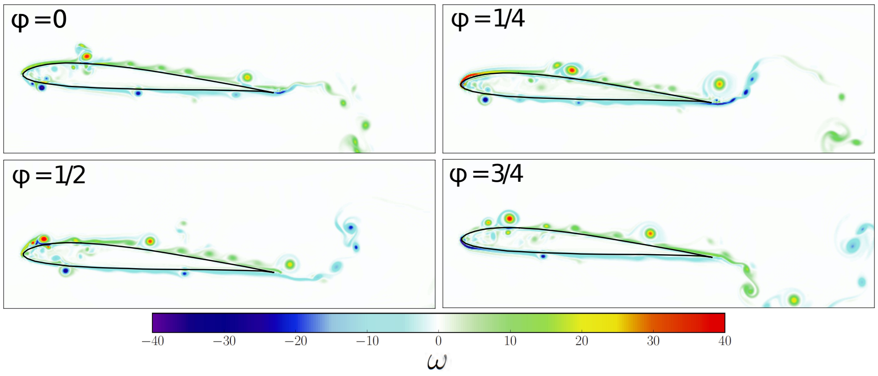
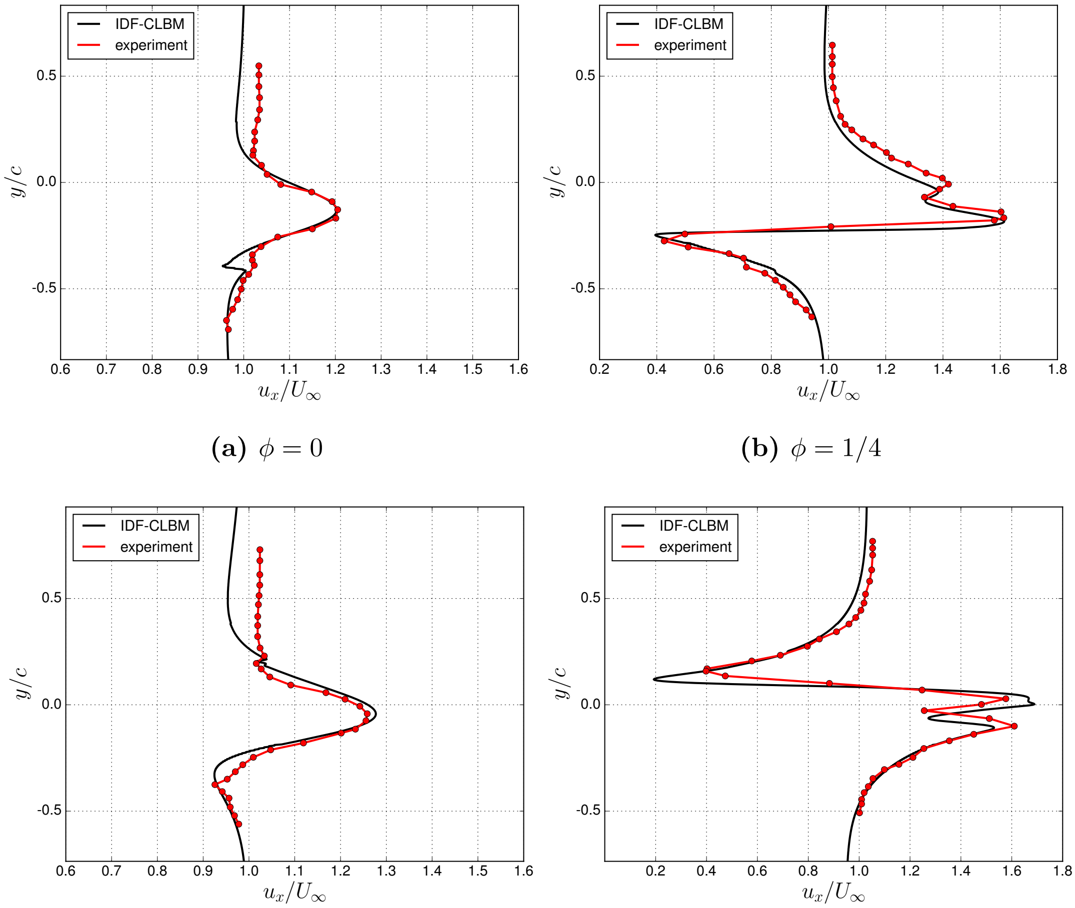


Figure 7.11: Vorticity field around a plunging SD7003 airfoil at $Re_c = 40000$ and four phases $\phi = 0, 1/4, 1/2, 3/4$.

shown in Fig. 7.11, $\phi = 3/4$, the vortex shed from the upper surface boundary layer is below the trailing edge vortex shed from the lower surface. A similar observation was made by Visbal [6]. This behaviour is consistent with the jet-like mean velocity profile observed at $\phi = 0, 1/2$, shown in Fig. 7.12a and Fig. 7.12c, where the airfoil experiences the maximum vertical displacements.

Fig. 7.13 shows the time history of the lift and drag coefficients. The results are compared with the 3-D computational results of Visbal [6]. This specific high-frequency low-amplitude motion of the SD7003 airfoil is expected to suppress the effect of the internal mass, computed with Eq. (3.37), in the computation of the aerodynamic coefficients. However, an overall 5.8% decrease, computed as shown in Section 7.5, and a phase difference of 1.6^o in the computed lift coefficient using the Lagrange point approximation is observed in Fig. 7.13b. The computed lift is found to be in good agreement both with the 3-D results of Visbal [6] and the prediction given by the inviscid theory [7]. Finally, the $C_L - C_D$ plot (Fig. 7.13a) over three consecutive cycle in the time-asymptotic state confirms the periodic nature of the aerodynamic forces.



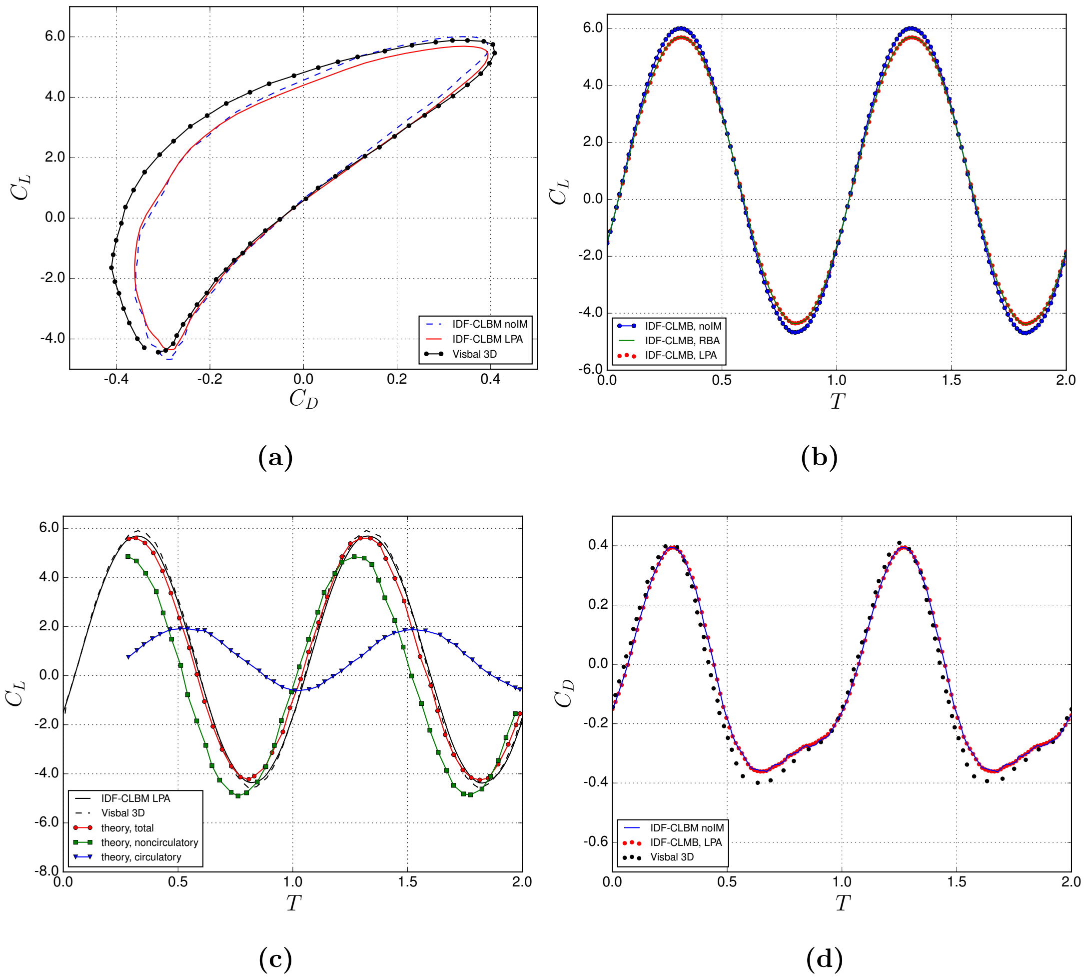
(a) $\phi = 0$ **(b)** $\phi = 1/4$

(c) $\phi = 1/2$ **(d)** $\phi = 3/4$

Figure 7.12: Velocity profiles for four phase angles (a) $\phi = 0$, (b) $\phi = 1/4$, (c) $\phi = 1/2$ and (d) $\phi = 3/4$ for $Re = 40000$ and $k = 3.93$. The experimental results of McGowan *et al.* [5] are used for comparison.

7.4 Plunging and flapping NACA-0014 airfoil

The effect of the chord-wise morphing on the aerodynamic coefficients of a NACA-0014 airfoil at Reynolds number $Re = 10^4$ for a range of prescribed plunging and flapping amplitudes is investigated in this section. For moderate Reynolds number flows, Tuncer and Kaya [151] argue that flapping-wing propulsion systems are in principle more efficient than their respective rotational propeller configurations.



(a) **(b)**

(c) **(d)**

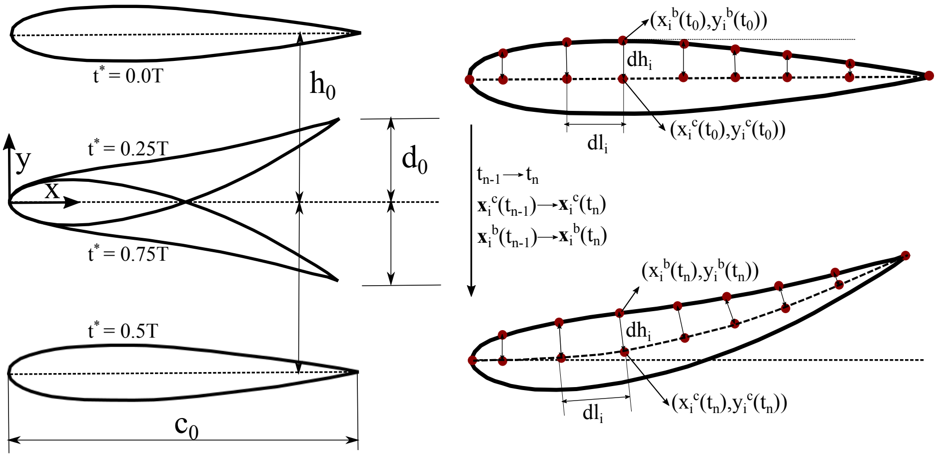
Figure 7.13: Computed lift and drag coefficients at $Re = 40000, k = 3.93$ and $\alpha_0 = 4^o$. The computational results of Visbal [6] and the theoretical inviscid results [7] are used for comparison.

The authors suggest that the aerodynamic loads of a flapping airfoil are strongly dominated by the leading-edge vortex dynamics and the trailing-edge vortex shedding. Murray and Howle [152] found that the aerodynamic efficiency of a flexible flapping thin foil may be significantly improved over its rigid counterpart. More recently, Miao and Ho [153] investigated the effect of flexure on the aerodynamic propulsive efficiency of a flapping flexible NACA-0014 airfoil and found that the overall propulsive efficiency is only improved for a very specific amplitude-deformation configuration. An interesting remark is the conclusion regarding the strong correlation between propulsive efficiency and reduced frequency of the oscillating motion. In this section, the flow parameters of Miao and Ho [153] are used. The airfoil undergoes a plunge motion with different flexure amplitude deformations as shown in Fig. 7.14a. The complex motion of the airfoil is expressed by

$$y_i(t) = -h_0\big(\cos(2\pi f(t-t_0)) + \cos(\pi)\big) - d_0\Big(\frac{x_i - x_0}{c_0}\Big)^2 \sin(2\pi f(t-t_0)), \quad (7.5)$$

where h_0 and a_0 denote the dimensionless amplitude of the oscillation and flexure amplitude respectively, f is the flapping frequency, x_0 is the x-coordinate of the leading edge and $c_0 = 300\delta x$ is the chord of the rigid airfoil. The foil in placed at the center of a $100c \times 100c$ computational domain. The inlet, outlet and free slip boundary conditions, as described in Chapter 2, have been used for the domain boundaries. A reduced frequency of $k = \omega c_0/U_\infty = 2$ is used in all cases, where $U_\infty = 0.02$. The reduced frequency $k = 2$ was set based on the observations of Miao and Ho [153] where they reported strong correlations between k and the propulsive efficiency of the foil. The first term in the right hand side of Eq. (7.5) denotes the plunge motion, $h(t)$, whereas the second term describes the deforming motion, $d(t)$, which is in $\psi = 90^o$ phase difference with the plunge motion in order to optimise the propulsive efficiency based on the observations of Miao and Ho [153] and Read *et al.* [154].

Let us first define the period averaged consumption power rate $\bar{P}$ and thrust force $\bar{F_x}$ as $\bar{P} = \frac{1}{T}\int_0^T F_n(t)\frac{dS_n}{dt}dt$ and $\bar{F_x} = \frac{1}{T}\int_0^T F_x(t)dt$, where $\frac{dS_n}{dt}$ is the deforming velocity. The propulsive efficiency n may now be expressed as $n = \xi/\delta$, where $\xi = \frac{\bar{F_x}}{(0.5\rho U_\infty^2 cs)} = \frac{1}{T}\int_0^T(-C_D)dt$ and $\delta = \frac{\bar{P}U_\infty}{(0.5\rho U_\infty^2 cs)}$ refer to the period-averaged thrust power and input power coefficients respectively. Effectively, optimisation of the propulsive efficiency is achieved by optimising the negative period averaged net drag forces. The authors reported that compared to a rigid



(a) **(b)**

Figure 7.14: (a) Schematic view of the wilfully deformed NACA-0014 airfoil and the flapping motion, (b) Updated surface coordinates under prescribed deforming motion of the NACA-0014 airfoil.

flapping foil configuration the propulsive efficiency of a wilfully deformable foil can be significantly optimised for $60^o < \psi < 150^o$ under certain flexure amplitudes.

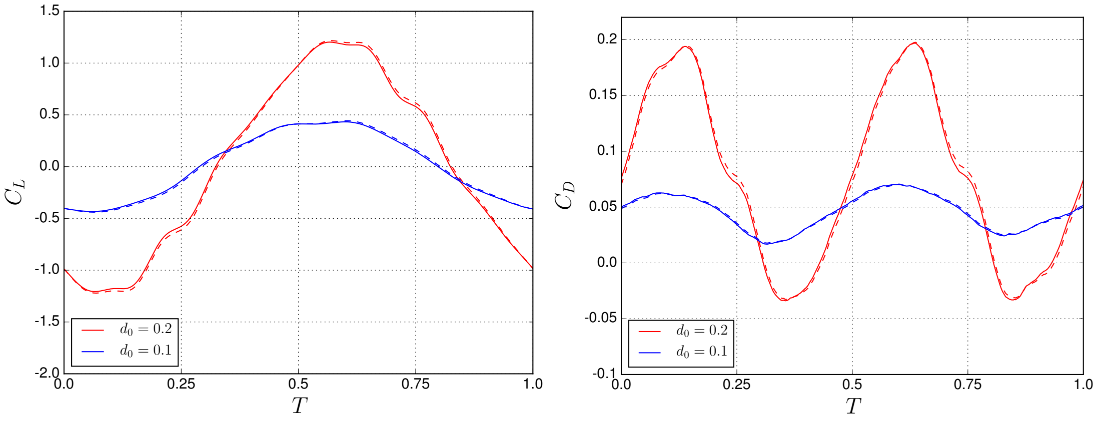
What differs from the work of Miao and Ho [153] is the deformation of each point of the surface in the x-direction, $x_i^b(t_n)$. In particular, assuming a constant chord length under any flexible deformation, a three-step process is used to calculate the surface deformation of the NACA-0014 airfoil, as shown in Fig. 7.14b. Under symmetry conditions and assuming incompressibility of the foil under any deformation, all points on the upper surface of the foil are projected onto the chord line. The discretisation lengths $dh_i(t_0)$ and $dl_i(t_0)$ are kept constant under any deformation.

Algorithm 10 Computation of surface deformation

```
Apply Eq. (7.5) to the chord line and compute $y_i^c(t_n)$. Under the constraint $dl_i(t_n) = dl_i(t_0)$ compute $x_i^c(t_n)$ using the following equation.
    $$x_i^c(t_n) = x_{i-1}^c(t_n) + \sqrt{dl_i(t_0)^2 - (y_i^c(t_n) - y_{i-1}^c(t_n))^2}, \quad (7.6)$$
    where the superscript $c$ refers to the chord of the foil.
Compute the chord tangents from Eq. 7.5, $(dy_i(t)/dx)$, and then compute the wall normals for all points $i$ using
    $$\mathbf{n}_i^c(t_n) = [A_0(\sin(2\pi ft)/c^2)(x_i^c(t_n) - x_0^c(t_n))]^{-1}, \quad (7.7)$$
    where $\mathbf{n}_i^c(t_n) = dx/dy_i(t)$.
Under the incompressibility constraint $dh_i(t_n) = dh_i(t_0)$, compute the foil surface deformation $x_i^b(t_n), y_i^b(t_n)$ using
    $$y_i^b(t_n) = \mathbf{n}_i^c(t_n)\sqrt{dh_i(t_0)^2/(1 + \mathbf{n}_i^c(t_n)^2)} + y_i^c(t_n), \quad (7.8)$$
    $$x_i^b(t_n) = (y_i^b(t_n) - y_i^c(t_n))/(\mathbf{n}_i^c(t_n)) + x_i^c(t_n), \quad (7.9)$$
    where the superscript $b$ refers to the surface of the foil.
```

7.4.1 Case 1 and 2: Deforming amplitudes $d_0 = 0.1c_0$ and $d_0 = 0.2c_0$

A chord-wise deforming motion with $d_0 = 0.1c_0$ and $d_0 = 0.2c_0$ is prescribed where no flapping motion is imposed to the foil. Fig. 7.15 shows the time evolution of the lift and drag coefficients under one period of oscillation. Since the overall motion of the foil is small, the effect of the internal mass does not significantly affect the computed aerodynamic coefficients. As shown in Fig. 7.15a much higher values



(a) **(b)**

Figure 7.15: Lift and Drag coefficients for rigid and flexible foil at plunging amplitude $h_0 = 0.0c_0$. The solid and dashed lines refer to the results with and without the LPA respectively.

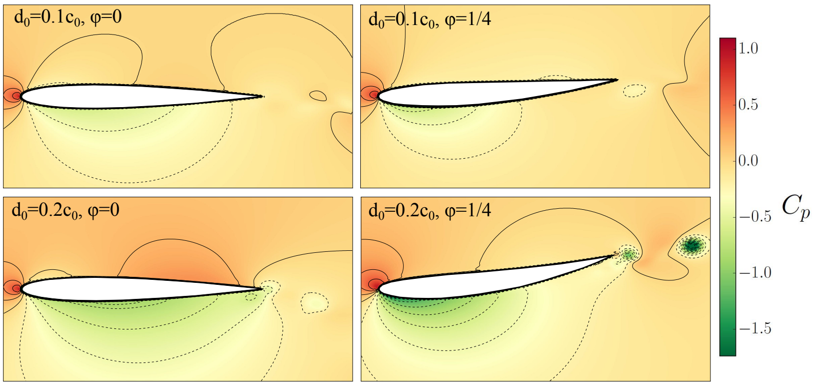
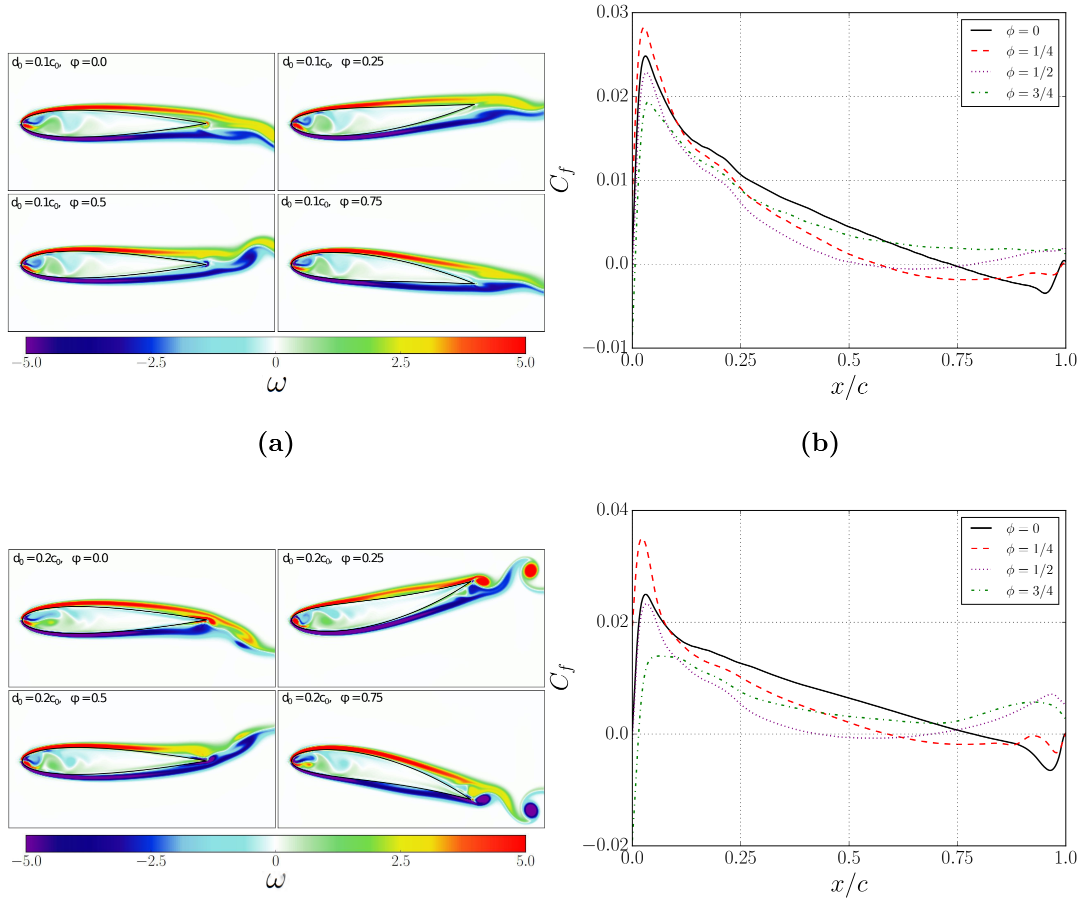


Figure 7.16: Pressure coefficient contour around a wilfully deformed NACA-0014 airfoil at $Re = 10000$, $d_0 = 0.1c_0, 0.2c_0$ and phase angles $\phi = 0, 1/4$.

of the lift coefficient are observed for $d_0 = 0.2c_0$. The stronger pressure gradient acting on the foil at a deformation amplitude $d_0 = 0.2c_0$, is showed in Fig. 7.16.

The computed time-averaged drag coefficient is $\overline{C}_D = 0.0557$ for the rigid NACA-0014 airfoil and $\overline{C}_D = 0.0457$ and $\overline{C}_D = 0.0763$ for the deformation amplitudes $d_0 = 0.1c_0$ and $d_0 = 0.2c_0$ respectively. The amplitude of the oscillation in the drag coefficient is much higher for $d_0 = 0.2c_0$. As shown in Fig. 7.17, between phases

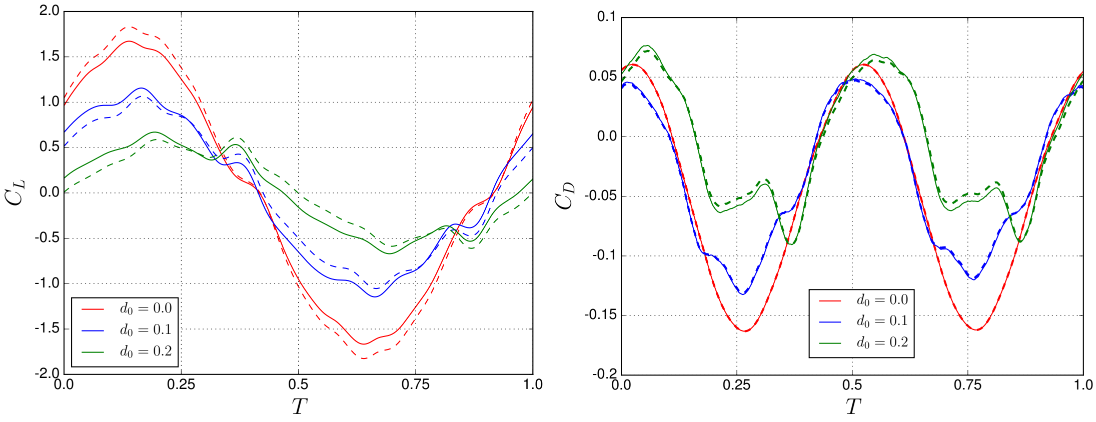
$\phi = 0$ and $\phi = 0.25$ the vorticity profiles around the airfoil differ significantly. For the larger deformation amplitude case, the vorticity generated at the trailing edge rolls over into smaller vortices that are shed downstream. Overall, under small deforming amplitudes, no significant improvement in the time-averaged thrust generation is observed.



(a) **(b)**

(c) **(d)**

Figure 7.17: Vorticity field around a wilfully deformed NACA-0014 airfoil at $Re = 10000$ for (a) $d_0 = 0.1c_0$ and (c) $d_0 = 0.2c_0$. Skin friction coefficient on lower airfoil surface at phases $\phi = 0, 1/4, 1/2, 3/4$ for deforming amplitudes (b) $d_0 = 0.1c_0$ and (d) $d_0 = 0.1c_0$.



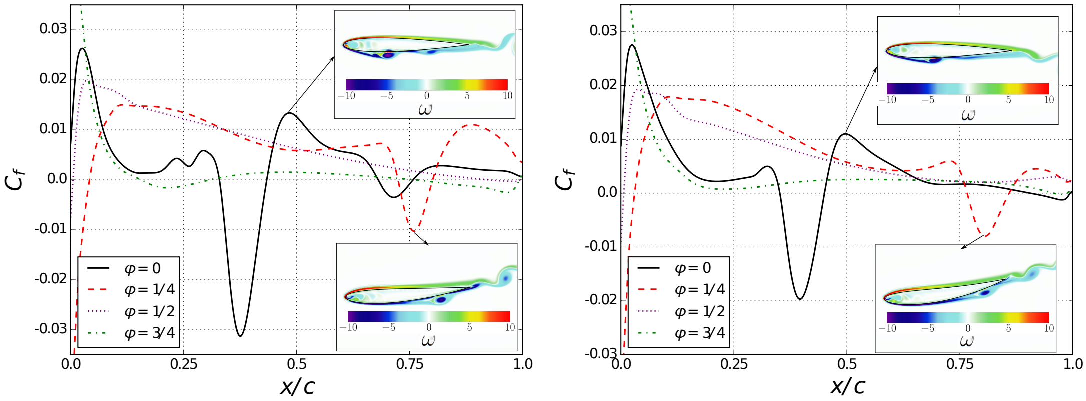
(a) **(b)**

Figure 7.18: Lift and Drag coefficients for rigid and wilfully deformed NACA-0014 airfoil at plunging amplitude $h_0 = 0.2c_0$ and deforming amplitudes $\alpha_0 = 0.0c_0, 0.1c_0, 0.2c_0$. The solid and dashed lines refer to the results with and without the LPA respectively.

7.4.2 Cases 3, 4 and 5: Plunging amplitude $h_0 = 0.2c_0$ and deforming amplitudes $d_0 = 0.1c_0, 0.2c_0$

The flow around a flapping and deforming NACA-0014 airfoil at $Re = 10000$ is investigated in this section. The prescribed flapping amplitude is $h_0 = 0.2c_0$.

We examine the cases of a rigid flapping foil and a deforming flapping foil with flexure amplitudes $d_0 = 0.1c_0$ and $d_0 = 0.2c_0$. Fig. 7.18 shows the time evolution of the lift and drag coefficients under one period of oscillation. The effect of the internal mass in the computation of the aerodynamic coefficients is significant, especially in the direction of the flapping motion. For $d_0 = 0.0c_0$ the internal mass corrections are stronger around the regions of the extremes, Fig. 7.18a. However, under the deforming motions, the corrections in the lift coefficient are important throughout the entire period of oscillation. It is shown that the present scheme can very accurately capture the momentum of the internal fluid even in



(a) **(b)**

Figure 7.19: Skin friction coefficient on lower airfoil surface at phases $\phi = 0, 1/4, 1/2, 3/4$ for plunging amplitude $h_0 = 0.2c_0$ and deforming amplitudes (a) $d_0 = 0.1c_0$ and (b) $d_0 = 0.2c_0$.

very complex motions. It is observed that as the flexure amplitude d_0 increases the extremes in the lift coefficient decrease. In the higher deforming amplitude case, $d_0 = 0.2c$, the overall periodic profile of both the lift and drag coefficients significantly differs from that of the rigid flapping foil. As shown in Fig. 7.18a, two peaks of approximately the same magnitude are observed between $t^* = 0$ and $0.5T$ and $t^* = 0.5T$ and $1.0T$. Similar observations can be made for the drag coefficient, Fig. 7.18b. The aerodynamic loads at this specific medium-amplitude low-frequency configuration are shown to be strongly dominated by the leading-edge vortex dynamics and the trailing-edge vortex shedding. Fig. 7.19 shows the skin friction coefficient and vorticity snapshots around the foil at phases $\phi = 0, 1/4, 1/2, 3/4$ and deforming amplitudes $d_0 = 0.1c_0$ and $d_0 = 0.2_0$.

In both cases, the position of the leading-edge vortex at $\phi = 0$ is approximately the same. However, for $d_0 = 0.1c_0$ a smaller amplitude vortex is located at 72% of the chord length leading to a trailing-edge vortex at $\phi = 0.25$. For the rigid plunging NACA-0014 aifoil, the period-average computed drag coefficient is $\overline{C}_D = -0.053$. The period-averaged drag force for the investigated deformation amplitudes $d_0 = 0.1c_0$ and $d_0 = 0.2c_0$ is $\overline{C}_D = -0.036$ and $\overline{C}_D = -0.007$ respectively. The period-averaged thrust force decreases from that of a rigid foil.

7.4.3 Case 6, 7 and 8: Plunging amplitude $h_0 = 0.4c_0$ and deforming amplitudes $d_0 = 0.1c_0, 0.3c_0$

The last cases in this section investigate the flow around a moving NACA-0014 airfoil at $Re = 10000$ with plunging amplitude $h_0 = 0.4c_0$ and deforming amplitudes $d_0 = 0.0c_0$, $d_0 = 0.1c_0$ and $d_0 = 0.3c_0$. The time evolution of the lift and drag coefficients under one period of oscillation for the three investigated cases are shown in Fig. 7.20. In this high-amplitude low-frequency motion configurations the internal mass effects are quite significant in the computation of the aerodynamic coefficients. The present LPA scheme can accurately capture the internal mass effects. Overall, as shown in Fig. 7.20a, the magnitude of the lift coefficient decreases drastically as the flexure amplitude d_0 increases. The difference in the lift force generation can be partially explained by investigating the pressure gradient acting on the foil at different phases of the oscillation.

As shown in Fig. 7.21, the leading-edge vortex formation and the subsequent trailing-edge vortex shedding dynamically affect the pressure around the foil. For the high deformation amplitude case, a significant observable difference in the trend of the lift coefficient over the period is observed. Similar observations can be made for the periodic behaviour of the drag coefficient, Fig. 7.20b. Overall, an improvement in the period-averaged thrust generation is observed for $d_0 = 0.1c_0$ where $\overline{C}_D = -0.249$ with respect to the rigid flapping foil case where $\overline{C}_D = -0.242$. The aerodynamic loads of the NACA-0014 airfoil at this specific high-amplitude low-frequency configuration are shown to be strongly dominated by the leading-edge vortex dynamics and the trailing-edge vortex shedding.

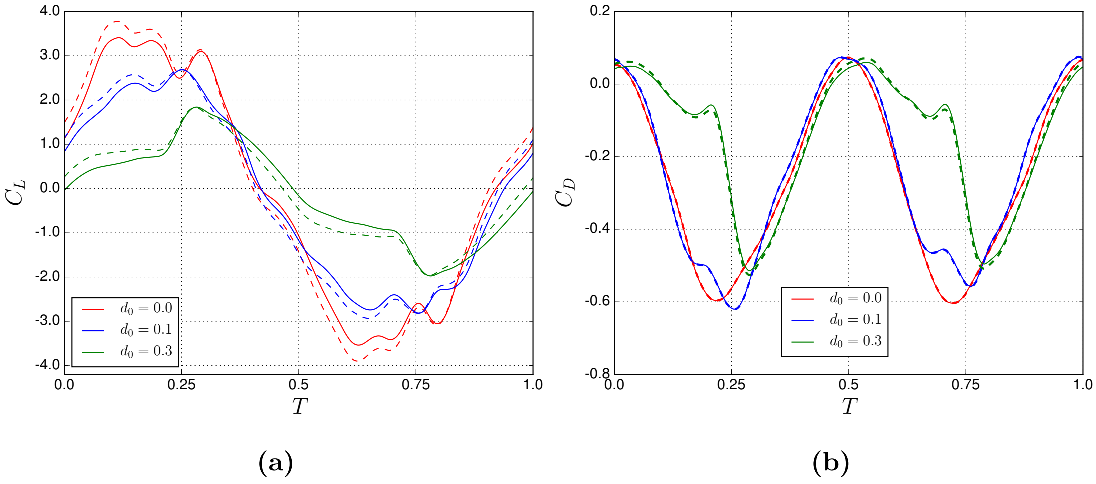


Figure 7.20: Lift and Drag coefficients for rigid and wilfully deformed NACA-0014 airfoil at plunging amplitude $h_0 = 0.4c_0$ and deforming amplitudes $\alpha_0 = 0.0c_0, 0.1c_0, 0.3c_0$. The solid and dashed lines refer to the results with and without the LPA respectively.

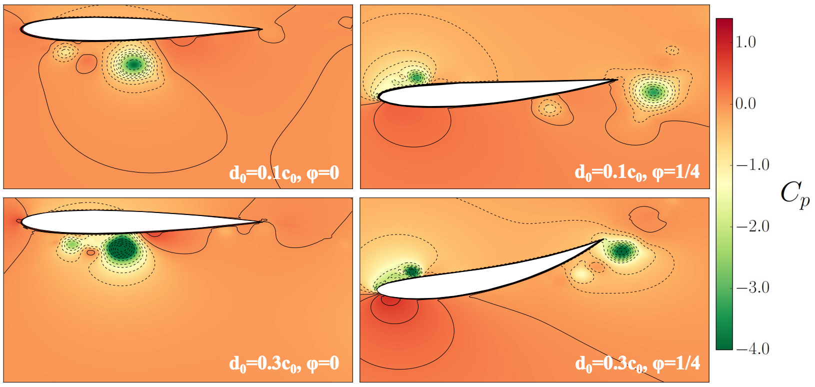
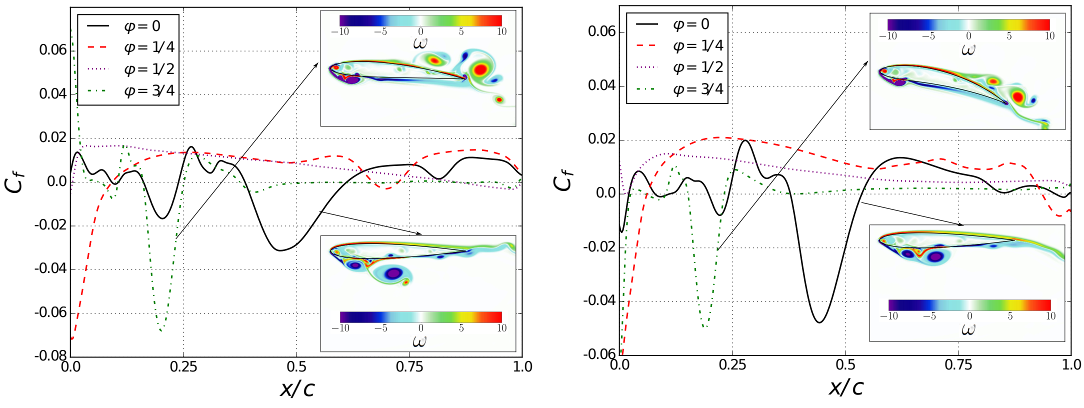


Figure 7.21: Pressure coefficient contour around a plunging wilfully deformed NACA-0014 airfoil at $Re = 10000$, $h = 0.4c_0$, $d_0 = 0.1c_0, 0.3c_0$ and phase angles $\phi = 0, 1/4$.

Fig. 7.22 shows the skin friction coefficient and vorticity snapshots around the foil at phases $\phi = 0, 1/4, 1/2, 3/4$ and deforming amplitudes $d_0 = 0.1c_0$ and $d_0 = 0.3c_0$. In both cases, the position of the two leading-edge vortices at $\phi = 0$ is approximately the same. However, the structure of the vortices differs



(a) (b)

Figure 7.22: Skin friction coefficient on lower airfoil surface at phases $\phi = 0, 1/4, 1/2, 3/4$ for plunging amplitude $h_0 = 0.4c_0$ and deforming amplitudes (a) $d_0 = 0.1c_0$ and (b) $d_0 = 0.3c_0$.

significantly leading to subsequently different vortex shedding dynamics. As shown in Fig. 7.22b, at $\phi = 3/4$ and $d_0 = 0.3c_0$ the structure of the trailing-edge vortices is more coherent and both vortices are closer to the surface of the foil.

It has been shown that the Lagrangian point approximation method can accurately capture the effects of the internal mass under various plunging-deforming configurations including both linear and angular motions. The effect of the deformation amplitude on the computation of the aerodynamic forces has been investigated and shown that the leading-edge vortex formation and the subsequent trailing-edge vortex shedding dominantly affect both the magnitude and the trend of the aerodynamic loads in all investigated plunging amplitudes $h_0 = 0.0c_0 - 0.4c_0$ and flexure amplitudes $d_0 = 0.0c_0 - 0.3c_0$. For zero plunging amplitude motions, an increase in both the lift and drag forces has been found. However, in the combined plunging-deforming motions, different aerodynamic behaviours have been observed. The formation of the leading-edge vortex seems to be mainly affected by the plunging amplitude, whereas the vortex transition along the surface of the foil and the respective thrust-indicative wake structure are strongly influenced by the deforming amplitude. However, in order to generalise the deformation effects, additional configurations must be examined. That includes the effect of the phase angle between the plunging and deforming motion, the effect of the reduced frequency and the effect of the Reynolds number.

7.5 Numerical accuracy of the Lagrange point approximation method

In this section the computational efficiency and accuracy of the Lagrangian point approximation method is investigated in order to define an optimum number of internal points and accurately represent the effects of the internal mass in the computation of the aerodynamic coefficients. First, consider an immersed body with a surface area A_Ω and define a number of internal Lagrangian points N_{int} with volume ΔV_{int}. As described in Section 2.3.2, the scaling factor is defined as $S_c = A_\Omega / N_{int}$. In this section, the accuracy of the LPA will be investigated for $S_c = 1.0$, $S_c = 2.0$ and $S_c = 4.0$. In all cases the internal Lagrangian points are equally spaced inside A_Ω. In order to quantify the effect of the LPA for the different scaling factors we use a modified normalised percentage error calculation procedure, shown below.

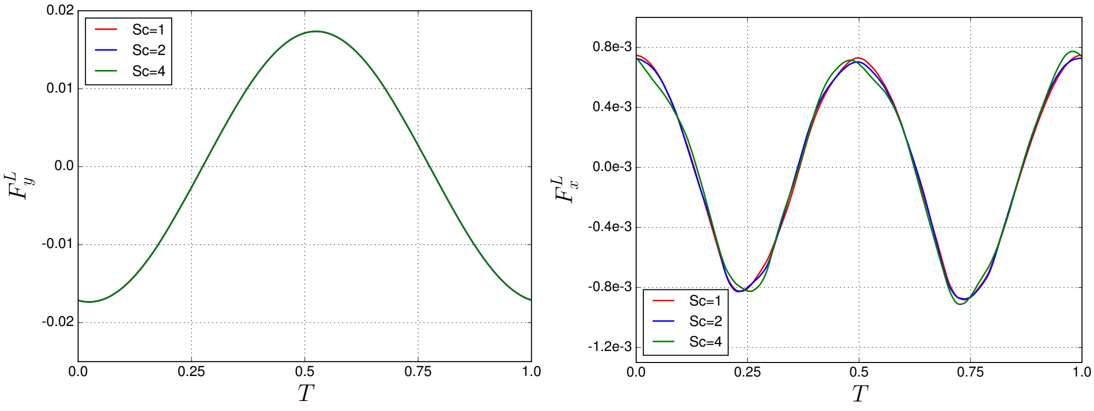
Consider two periodic functions $f^{(1)}(t)$ and $f^{(2)}(t)$ and let $0.0T \leq t \leq 1.0T$, where T is the period of the oscillation. In order to quantify the observable percentage difference between $f^{(1)}(t)$ and $f^{(2)}(t)$ and avoid the regions where the values approximate the zero value, leading to big percentage errors, the following normalisation procedure is applied.

The normalised percentage errors of the lift and drag coefficients for all the cases investigated in this study are presented in Table 6.2. All errors are computed

Algorithm 11 Normalised percentage error between periodic functions

```
Compute the root mean square of $f^{(1)}(t)$ using
    $$RMS_{f^1} = \sqrt{\frac{1}{N}\sum_{1}^{N} f^{(1)}(t)^2}, \qquad (7.10)$$
    where $N$ is the total number of discrete points.
Compute the distance $D_{f^1}$ of $f^{(1)}(t)$ from its root mean square value for all $N$ as $D_{f^1} = RMS_{f^1} - f^{(1)}(t)$.
Shift both functions $f^{(1)}(t)$ and $f^{(2)}(t)$ using the computed distance $D_{f^1}$ as $f^{(1)}_{new}(t) = D_{f^1} + f^{(1)}(t)$ and $f^{(2)}_{new}(t) = D_{f^1} + f^{(2)}(t)$ respectively.
Compute the mean absolute percentage difference between $f^{(1)}_{new}(t)$ and $f^{(2)}_{new}(t)$ as
    $$E = \frac{1}{N}\frac{\mid f^{(1)}_{new}(t) - f^{(2)}_{new}(t) \mid}{f^{(1)}_{new}(t)}. \qquad (7.11)$$
```

with respect to the aerodynamic coefficients as computed without considering the effects of the internal mass.



(a) **(b)**

Figure 7.23: Time history of internal Lagrangian forces (F_y^L, F_x^L) for $S_c = 1, 2, 4$ at plunging amplitude $h_0 = 0.4c_0$ and deforming amplitude $d_0 = 0.3c_0$.

Table 7.2: Numerical accuracy of the Lagrange point approximation method on the lift coefficient C_L for three scaling factors $S_c = 1.0$, $S_c = 2.0$ and $S_c = 4.0$. All numbers are expressed as % differences.

Case	Configuration	C_L		
		$E_N^{Sc=1}$	$E_N^{Sc=2}$	$E_N^{Sc=4}$
Cylinder	$A_m = 2.5D/\pi$	-	-	-
Pitching foil	$A_D = 0.36D$	7.974	7.970	7.961
	$A_D = 0.71D$	7.098	7.094	7.086
	$A_D = 1.07D$	6.250	6.247	6.239
	$A_D = 1.77D$	5.165	5.162	5.157
SD7003	$h_0 = 0.05c$	5.435	5.433	5.428
NACA-0014	$h_0 = 0.0c_0, \quad d_0 = 0.1c_0$	3.568	3.567	3.567
	$h_0 = 0.0c_0, \quad d_0 = 0.2c_0$	2.573	2.574	2.574
	$h_0 = 0.2c_0, \quad d_0 = 0.0c_0$	7.739	7.741	7.739
	$h_0 = 0.2c_0, \quad d_0 = 0.1c_0$	13.947	13.945	13.952
	$h_0 = 0.2c_0, \quad d_0 = 0.2c_0$	25.270	25.266	25.277
	$h_0 = 0.4c_0, \quad d_0 = 0.0c_0$	10.499	10.498	10.500
	$h_0 = 0.4c_0, \quad d_0 = 0.1c_0$	12.634	12.633	12.635
	$h_0 = 0.4c_0, \quad d_0 = 0.3c_0$	18.332	18.330	18.334

From Tables 7.2 and 7.3 it is clear that the accuracy of the LPA is not significantly affected when 25%, ($S_c = 4$), of the initial internal Lagrangian points is used. It should be noted however, that the initial number of Lagrangian points N_{int}, ($S_c = 1$), should be big enough in order to accurately capture the internal mass effects. There are effects however, that cannot be directly extracted from the normalised percentage differences. As shown in Fig. 7.23b, the internal forces differ both in magnitude and in phase when only a part of the internal Lagrangian points is used. In the present test cases where the motion of the body is prescribed those differences are negligible. However, in cases where the motion of the immersed body is affected by the fluid forces acting on it, a more careful consideration regarding the number of Lagrangian points is required. In Tables 7.4 and 7.5, the maximum absolute percentage differences on the lift and drag

Table 7.3: Numerical accuracy of the Lagrange point approximation method on the drag coefficient $C_D$for three scaling factors $S_c = 1.0$, $S_c = 2.0$ and $S_c = 4.0$. All numbers are expressed as $\%$ differences.

Case	Configuration	C_D		
		$E_N^{Sc=1}$	$E_N^{Sc=2}$	$E_N^{Sc=4}$
Cylinder	$A_m = 2.5D/\pi$	34.051	34.049	34.048
Pitching foil	$A_D = 0.36D$	1.466	1.471	1.470
	$A_D = 0.71D$	6.108	6.106	6.090
	$A_D = 1.07D$	10.278	10.272	10.257
	$A_D = 1.77D$	10.954	10.948	10.925
SD7003	$h_0 = 0.05c$	1.084	0.992	1.114
NACA-0014	$h_0 = 0.0c_0, \quad d_0 = 0.1c_0$	1.859	1.867	1.864
	$h_0 = 0.0c_0, \quad d_0 = 0.2c_0$	3.263	3.270	3.267
	$h_0 = 0.2c_0, \quad d_0 = 0.0c_0$	0.157	0.225	0.294
	$h_0 = 0.2c_0, \quad d_0 = 0.1c_0$	1.348	1.403	1.453
	$h_0 = 0.2c_0, \quad d_0 = 0.2c_0$	7.371	7.372	7.318
	$h_0 = 0.4c_0, \quad d_0 = 0.0c_0$	0.127	0.182	0.206
	$h_0 = 0.4c_0, \quad d_0 = 0.1c_0$	2.993	3.063	3.247
	$h_0 = 0.4c_0, \quad d_0 = 0.3c_0$	3.812	3.807	3.828

coefficients for the flow around the NACA-0014 airfoil with $h_0 = 0.4c_0$ is shown. A similar behaviour is observed, indicating that for the cases studied in this paper it is sufficient to represent the effects of the internal mass using 25% of N_{int}.

Table 7.4: Maximum absolute percentage difference on the computed lift coefficient C_L for three scaling factors $S_c = 1.0$, $S_c = 2.0$ and $S_c = 4.0$.

Case	Configuration	C_L		
		$E_m^{Sc=1}$	$E_m^{Sc=2}$	$E_m^{Sc=4}$
NACA-0014	$h_0 = 0.4c_0, \quad d_0 = 0.0c_0$	11.751	11.748	11.751
	$h_0 = 0.4c_0, \quad d_0 = 0.1c_0$	15.025	15.022	15.031
	$h_0 = 0.4c_0, \quad d_0 = 0.3c_0$	28.356	28.352	28.361

Table 7.5: Maximum absolute percentage difference on the computed drag coefficient C_D for three scaling factors $S_c = 1.0$, $S_c = 2.0$ and $S_c = 4.0$.

Case	Configuration	C_D		
		$E_m^{Sc=1}$	$E_m^{Sc=2}$	$E_m^{Sc=4}$
NACA-0014	$h_0 = 0.4c_0, \quad d_0 = 0.0c_0$	0.475	0.609	0.820
	$h_0 = 0.4c_0, \quad d_0 = 0.1c_0$	1.513	1.506	1.523
	$h_0 = 0.4c_0, \quad d_0 = 0.3c_0$	6.860	6.783	7.231

Chapter 8

Hybrid higher-order IBM - Computational results

In this chapter, a novel hybrid higher-order immersed boundary method is presented in order to address the two major drawbacks of the conventional IBM as described in Chapter 5, the velocity gradient discontinuities and the pressure correction on the boundary surface. A detailed description of the hybrid algorithm can be found in Chapter 5.

The force spreading operation in the IBM introduces a discontinuity on the velocity gradient across the boundary leading to only first-order accurate in space flow computations. This effect is strongly enhanced in high Reynolds number flows where very sharp boundary layer profiles are observed, leading to unstable boundary conditions. The proposed method of smoothly reconstructing the boundary layer profile inside the solid domain effectively suppresses this discontinuity away from the boundary and increases the accuracy of the solution. The effect of the present scheme to higher Reynolds number flows should be further examined.

The proposed pressure correction algorithm (PCA) is derived from the Navier-Stokes equations and may effectively be considered as a way of prescribing Neumann boundary conditions in the immersed boundary framework. Investigation of the flow around stationary and moving boundaries demonstrate the drastic improvements on the pressure distribution across the boundary with the proposed scheme. The corrections in the pressure are achieved by reconstructing the populations near the boundary nodes using extrapolated density and non-equilibrium distribution values. The proposed PCA is very flexible and can be implemented to any IBM. The improvements in the pressure distribution would be of great interest in fully coupled fluid structure interaction applications.

The resulting coupled immersed boundary lattice - Boltzmann solver has been applied to viscous flows around a stationary cylinder, an in-line oscillating cylinder, an oscillating cylinder in a cross flow and an elliptical wing undergoing a hovering flight. Good agreement with experimental results and other numerical schemes has been obtained. Significant improvements in the computation of the aerodynamic coefficients at moving boundary applications is achieved. It is demonstrated that the proposed hybrid solver successfully suppresses the velocity gradient discontinuities and corrects the pressure distribution along the boundary surface.

8.1 2D Taylor - Couette flow

First, the accuracy of the proposed hybrid scheme is examined by investigating the Taylor - Couette flow between two concentric circular cylinders. The flow is induced by rotating the inner cylinder around the axial z direction. Let R_1 and $R_2 = 2R_1$ be the radius of the inner and outer cylinder respectively. The two cylinders are located on a $N \times N$ domain on the $x-y$ plane normal to the z axis. The domain size N is chosen as $N = 2[R_2 + (N_g + 5)\delta x]$, where N_g is the number of ghost nodes placed on the outer cylinder. The Reynolds number is defined as $Re = U_{max}R_2/\nu = 10$, where U_{max} is the maximum rotating speed of the inner cylinder. Similar to the work of Suzuki and Inamuro [130] the relaxation time is set at $\tau = 0.68$. Periodic boundary conditions have been used for the domain boundaries.

The angular speed of the inner cylinder is given by

$$\Omega_{C1}(t) = \frac{1}{2}\left[1 + \tanh\left(\frac{U_{max}t - 0.4R_1}{0.1R_1}\right)\right]\frac{U_{max}}{R_1}, \tag{8.1}$$

where $\Omega_{C1}(t)$ tends to U_{max}/R_1 as $t \to \infty$. At $t = 0$ the fluid is at rest. The analytical steady state solution for the flow is given by

$$u_\theta(r) = -\frac{R_1U_{max}}{R_2^2 - R_1^2}\left[1 - \left(\frac{R_2}{r}\right)^2\right]r, \tag{8.2}$$

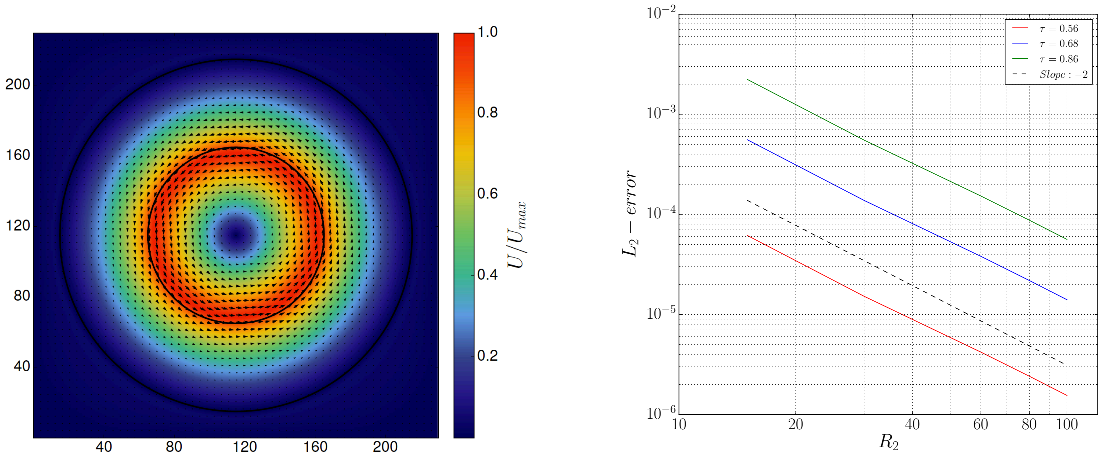
$$p(r) = \left(\frac{R_1U_{max}}{R_2^2 - R_1^2}\right)^2\left[\frac{r^2}{2} - \frac{R_2^4}{2r^2} - 2R_2^2\ln\left(\frac{r}{R_2}\right)\right] + p_0, \tag{8.3}$$

where p_0 is a reference pressure, r is the distance the center of rotation and $u_\theta(r)$ is the azimuthal flow velocity. The global error of the velocities is evaluated when the flow has reached a steady state using the following L_2 norm error:

$$L_2^{error} \equiv \sqrt{\Big(\sum_{R_2/R_1}(u_x^c - u_x^a)^2 + (u_y^c - u_y^a)^2\Big)/N}, \tag{8.4}$$

where N is defined as $N = \pi(R_2^2 - R_1^2)$ and the superscripts, a and c, refer to the analytical and computational values respectively. $\sum_{R_2/R_1}$ means that the summation is made over lattice points in the range $R_1^2 \leq x^2 + y^2 \leq R_2^2$. The computed steady state normalised velocity field for $R_2 = 100$ and $N_g = 3$ is shown in Fig. 8.1a.

Fig. 8.1b shows the global L_2-error versus the number of grid points along

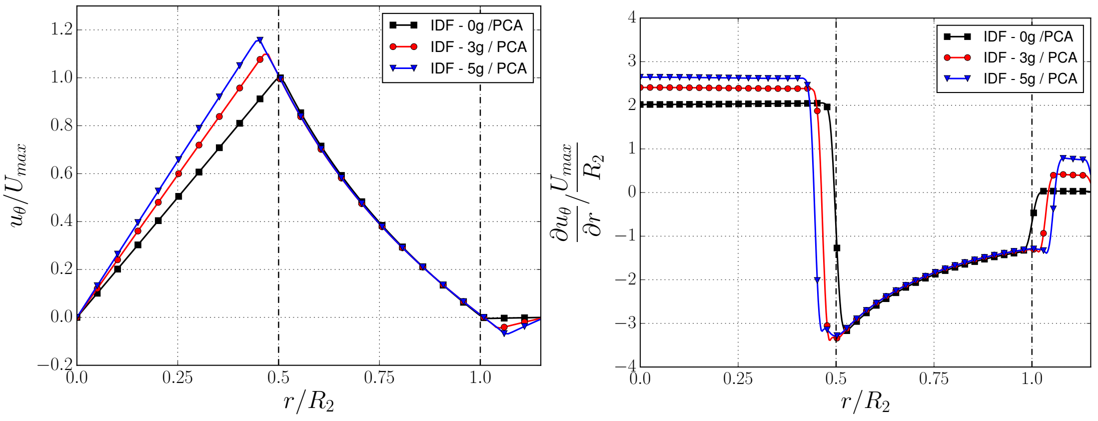


(a) **(b)**

Figure 8.1: (a) Steady state velocity solution. (b) Global accuracy of the present hybrid IDF-CLBM for $N_g = 3$ and PCA. The effect of three relaxation parameters $\tau = 0.56, 0.68, 0.86$ on the overall accuracy at $Re = 10$ is examined.

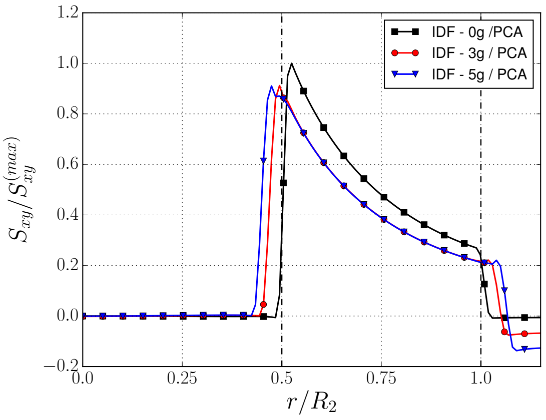
the outer cylinder for three relaxation times $\tau = 0.56, 0.68, 0.86$ using $N_g = 3$ and coupled with the PCA. Let us define the apparent order of convergence as $p = \ln\left((L_2^{i+1} - L_2^{i})/(L_2^{i} - L_2^{i-1})\right)/\ln(r)$, where r is the grid refinement ratio. The superscripts in the L_2 terms denote the refinement levels ($R_2 = 15, 30, 60, 80, 100$). For all three examined relaxation times, the apparent order of convergence is $p = 1.99$ and $p = 1.96$ for the IDF-CLBM with $N_g = 3$ between the refining regions $R_2 = 30 - 100$ and $R_2 = 15 - 30$ respectively.

The velocity and the velocity gradient across the boundary using the expanded IDF-CLBM with $N_g = 0, 3, 5$ and $R_2 = 100$ coupled with the PCA, are shown in Fig. 8.2. For the IDF-CLBM with $N_g = 0$, while the velocity profile is continuous across the boundary, the gradient of the velocity is discontinuous, as shown in Fig. 8.2b at $r/R_2 = 0.5$ and $r/R_2 = 1.0$. Similar results have been reported by Suzuki and Inamuro [130] for $N_g = 0$. For $N_g = 3, 5$ the velocity profile is smoothly expanded across the boundary into the body domain. The discontinuity is shifted inside the body domain away from the computational area of interest.



(a) **(b)**

Figure 8.2: (a) Velocity profile and (b) velocity gradient profile at the cross section $x = x_c$ at $Re = 10$.



(a)

Figure 8.3: Stress S_{xy} profile at the cross section $x = x_c$ at $Re = 10$.

Similar observation can be made by looking at the stress distribution in Fig. 8.3. The stresses are computed using the non-equilibrium distribution parts.

Finally, the effect of the pressure correction algorithm on the computed pressure is presented in Table 8.1. As shown, the PCA effect is stronger for $N_g = 0$, where the L_2-error is decreased by an order of magnitude.

Table 8.1: Effect of the PCA algorithm on the accuracy of the pressure computation for $R_2 = 100$ and $N_g = 0, 1, 3$ at $Re = 10$.

Case	$L_2(P)$, no PCA	$L_2(P)$, PCA
$N_g = 0$	$3.58 \cdot 10^{-7}$	$3.47 \cdot 10^{-8}$
$N_g = 1$	$1.79 \cdot 10^{-7}$	$6.71 \cdot 10^{-8}$
$N_g = 3$	$7.04 \cdot 10^{-8}$	$4.81 \cdot 10^{-8}$

8.2 Flow around a circular cylinder

Similar to Chapter 6, the flow past a stationary circular cylinder is examined in this section using the hybrid higher-order scheme described in Chapter 5. In this extensively studied case, the flow behaviour changes according to the Reynolds number which is defined as $Re = U_\infty D/\nu$, where D is the diameter of the cylinder, U_∞ is the freestream velocity and ν is the kinematic viscosity. The steady and unsteady flows, at $Re = 40$ and $Re = 100$ respectively, will be examined. In the present scheme the boundary forces are computing as

$$F_{\mathscr{L},\beta} = -\sum_{i,j} f_\beta(x_i, y_j)h^2, \tag{8.5}$$

where the summation is performed over the Eulerian forces $f_\beta(x_i, y_j)$ and h is the grid spacing. The aerodynamic coefficients are computed using Eqns. (6.6 - 6.7). The pressure coefficient may be computed as

$$C_p = \frac{p(\mathbf{X}_l) - p_\infty}{0.5\rho U_\infty^2}, \tag{8.6}$$

where p_∞ is the freestream pressure and $p(\mathbf{X}_l)$ is the interpolated pressure on the boundary nodes. The computational domain is initialised using $\rho_0 = 1.0$ and the freestream velocity is set to $U_\infty = 0.1$ and $U_\infty = 0.04$ for $Re = 40$ and $Re = 100$ respectively. The computational domain is $100D \times 100D$ with nine levels of grid refinement. The region around the cylinder is $4D \times 2D$ with a uniform mesh of 405×205 grid points. The inlet, outlet and free slip boundary conditions, as described in Chapter 2, have been used for the domain boundaries. Unless otherwise specified, the number of iterations in the immersed boundary algorithm for $N_g = 0, 1, 2, 3$ is set to $N = 10$ in the present study. A detailed comparison of the computational cost of the present higher-order IBM with pressure correction and the previous IDF IB scheme is presented in Appendix A.3.

8.2.1 Re=40

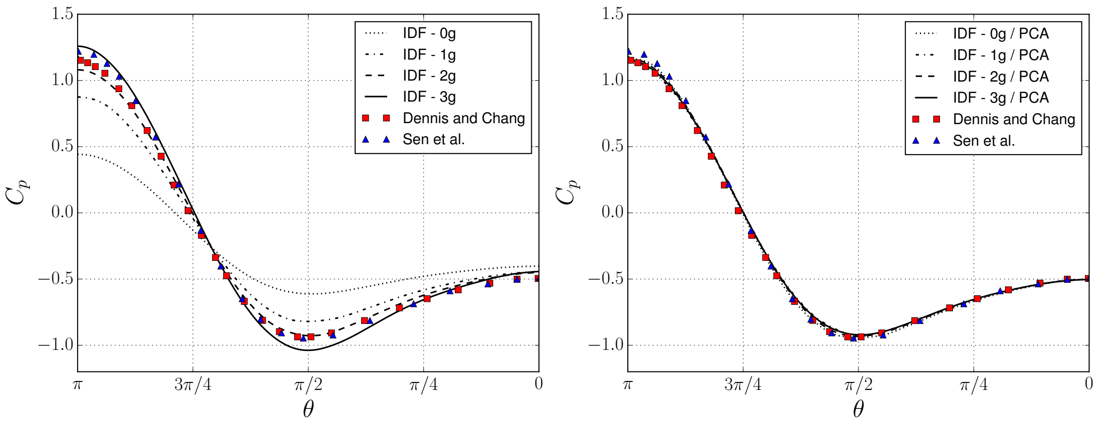
At $Re = 40$, a development of two symmetric, stationary recirculating eddies is observed behind the cylinder. The wake length L_w, or recirculation length, is defined as $L_w = 2L/D$, where L is the distance from the rearmost point of the cylinder to the end of the wake, as shown is Fig. 6.3. The separation angle θ_s is defined as the angle between the rearmost point of the cylinder and the point s on the cylinder surface where the shear stress is zero. The drag coefficient C_d, the wake length L_w and the separation angle θ_s are compared with other numerical and experimental results [134, 135, 104, 137, 55] in Table 8.2. Good overall agreement between the present scheme and previous works is observed.

The effect of the pressure correction algorithm in the computation of the pressure coefficient on the surface of the cylinder is investigated in Fig. 8.4. Fig. 8.4a compares the computed values of the pressure coefficient for $N_g = 0, 1, 2, 3$ with the experimental and numerical values presented in [155, 156]. It is observed that for $N_g = 0$ (conventional IBM), very low pressure values are obtained. However, the expansion of the velocity inside the immersed surface indirectly and somehow arbitrarily affects the pressure. This behaviour agrees with the observations of Suzuki [130].

However, as shown in Fig. 8.4b, very accurate pressure values are obtained with

Table 8.2: Comparison of Drag coefficient, wake length L_w and separation angle θ_s for steady flow over a circular cylinder at $Re = 40$.

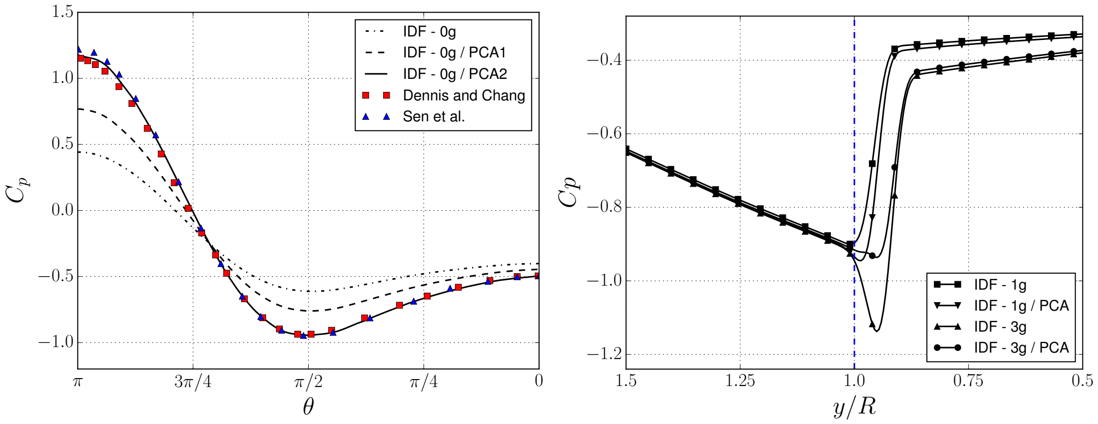
Case	References	C_d	L_w	θ_r (^{o})
$Re = 40$	[134] - Finite difference, NSE	1.522	4.69	53.8
	[135] - Body-fitted grid, NSE	1.498	4.48	-
	[104] - Implicit velocity correction, IBM	1.565	4.62	-
	[137] - Vorticity streamfunction, NSE	1.550	4.357	53.34
	[55] - Iterative direct forcing IBM, CLBM	1.524	4.61	53.12
	Present - $N_g = 0$, PCA IDF-CLBM	1.550	4.573	53.41
	Present - $N_g = 1$, PCA IDF-CLBM	1.528	4.622	53.19
	Present - $N_g = 2$, PCA IDF-CLBM	1.531	4.624	53.27
	Present - $N_g = 3$, PCA IDF-CLBM	1.531	4.624	53.28



(a) **(b)**

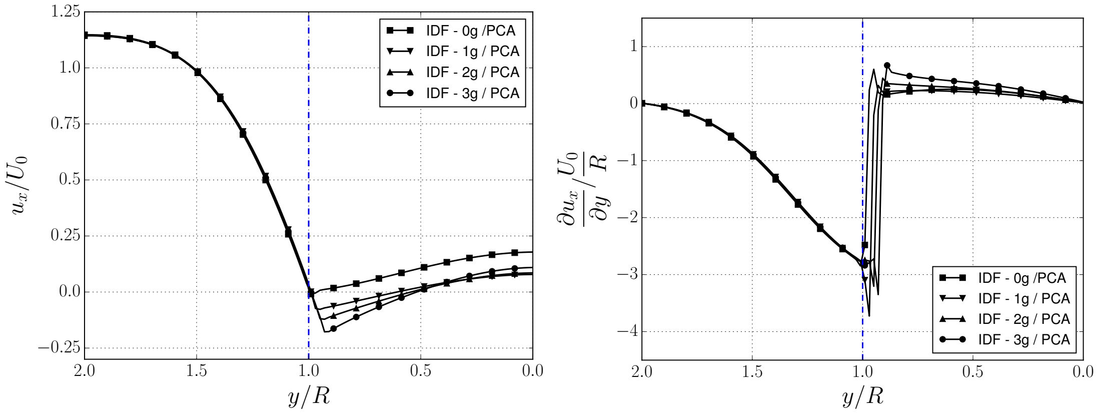
Figure 8.4: Pressure coefficient on the surface of the cylinder at $Re = 40$, (a) without PCA and (b) with PCA.

the pressure correction algorithm regardless of the number of ghost nodes used in the expanded IBM. In Fig. 8.5 the effect of incorporating the forces in the PCA scheme is presented. As seen, applying the Neumann boundary condition for the pressure $\partial p/\partial n = 0$ without considering the presence of the external forces (PCA1 scheme) is not sufficient to acquire correct pressure values on the surface of the body. However, when the proposed scheme (PCA2 scheme) is used, the correct values are obtained. Finally, as shown in Fig. 8.5b, the pressure correction algorithm with $N_g = 1, 3$ smoothly expands the pressure inside the immersed body.



(a) **(b)**

Figure 8.5: Pressure coefficient at $Re = 40$ (a) on the surface of the cylinder and (b) at the cross section $x = x_c$.



(a) **(b)**

Figure 8.6: (a) Velocity profile and (b) velocity gradient profile at the cross section $x = x_c$ at $Re = 40$.

Finally, Fig. 8.6 shows the expansion of the velocity profile at $x = x_c$, where

Table 8.3: Comparison of lift coefficient, drag coefficients and Stouhal number for unsteady flow over a circular cylinder at $Re = 100$.

Case	References	$\bar{C_d}$	C_l	St
$Re = 100$	[138] - Experiment	-	-	0.166
	[139] - Body-fitted method, NSE	1.35	± 0.339	0.164
	[104] - Implicit velocity correction IBM	1.364	± 0.344	0.163
	[76] - Higher order IBM-NSE	1.34	± 0.315	0.164
	[79] - Exterior sharp direct forcing LBM	1.336	± 0.329	0.165
	[55] - Iterative direct forcing IBM	1.334	± 0.331	0.165
	Present - $N_g = 0$ PCA	1.344	± 0.319	0.164
	Present - $N_g = 1$ PCA	1.325	± 0.327	0.164
	Present - $N_g = 2$ PCA	1.335	± 0.334	0.165
	Present - $N_g = 3$ PCA	1.334	± 0.333	0.165

(x_c, y_c) is the center of mass. It is observed that for $N_g = 2$ and $N_g = 3$ the velocity gradient profile is smoothly expanded inside the body domain.

8.2.2 Re=100

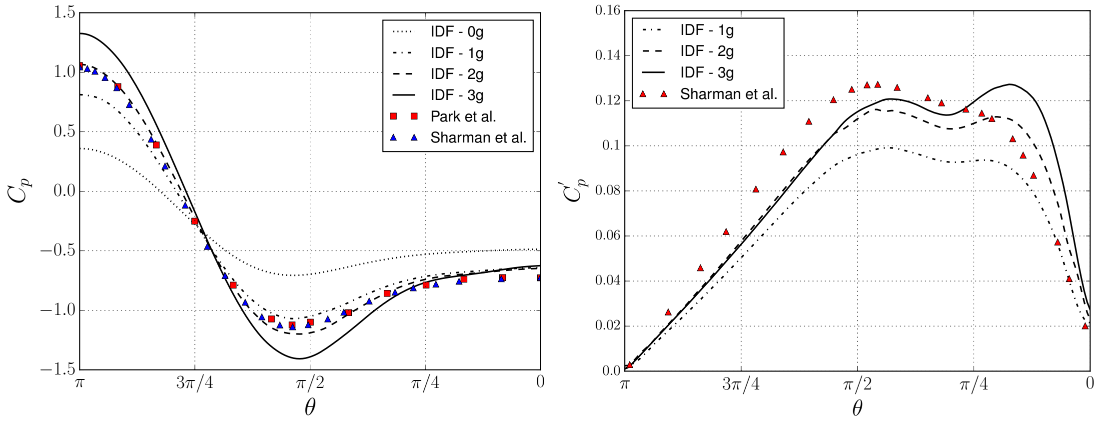
At $Re = 100$, vortices are shed from the body. The Strouhal number is defined as $St = f_d D/U_\infty$, where f_d is the shedding frequency. Table 8.3 presents the average drag coefficients, minimum and maximum values of the lift coefficients and the Strouhal numbers as computed with the present scheme. The results are compared with other numerical schemes and experiments[138, 139, 104, 76, 79, 55] and good agreement is observed.

Figures 8.7 and 8.8 show the average and root mean square values of the pressure coefficient obtained with $N_g = 0, 1, 2, 3$ without and with the pressure correction algorithm respectively. The root mean square values C_p^{RMS}, also referred to as

$C_p^{'}$ are computed as

$$C_p^{'} = \sqrt{\frac{\sum_{i=1}^{N}(C_p^i - C_p^{mean})^2}{N}}, \tag{8.7}$$

where C_p^{mean} is the period averaged pressure coefficient. It is observed that for $N_g = 3$ the compute average values are significantly higher that the ones reported in Park *et al.* [157] and Sharman *et al.* [158]. The computed RMS values without the PCA are not in good agreement with the expected values. However, accurate average and RMS pressure values are obtained when the PCA is coupled with the IBM, as shown in Fig. 8.8.

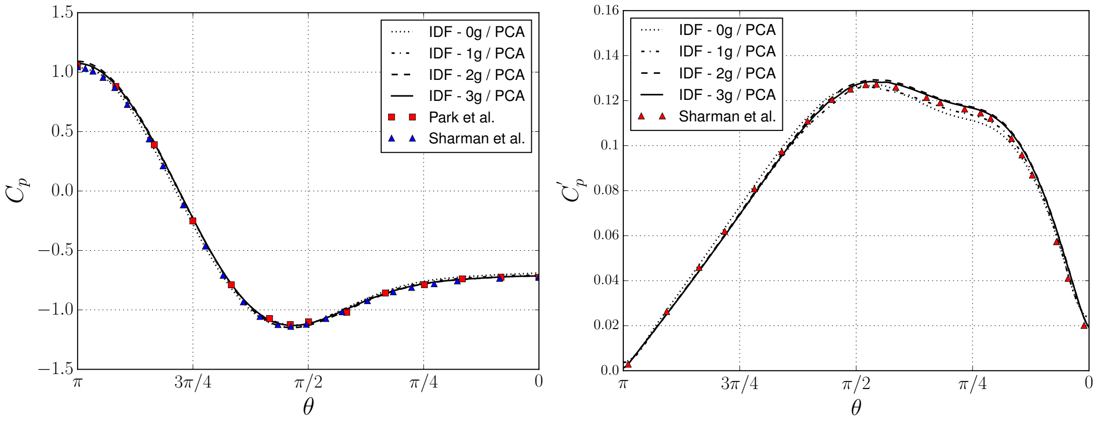


(a) **(b)**

Figure 8.7: (a) Average pressure coefficient and (b) root mean square pressure coefficient on the surface of the cylinder at $Re = 100$ without PCA.

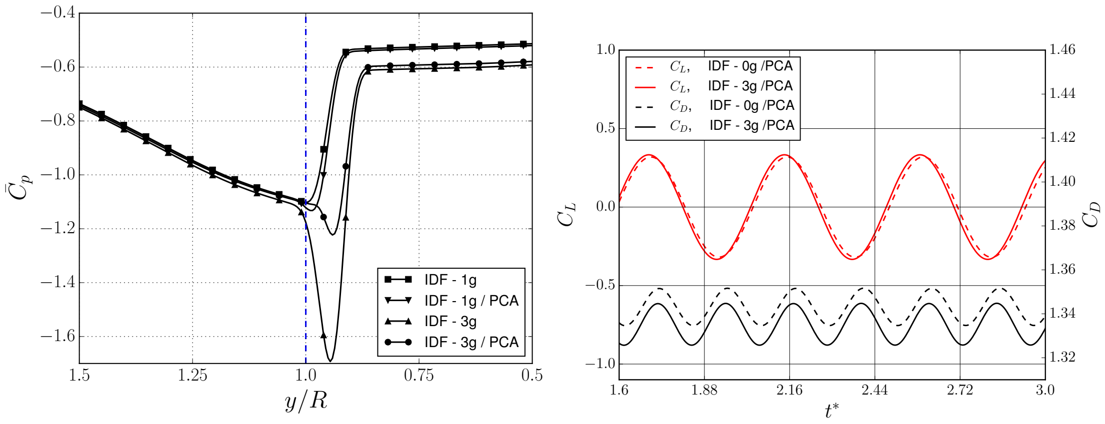
Fig. 8.9 shows the computed average pressure coefficient at the cross section of the cylinder at $x = x_c$ and the time evolution of the aerodynamic coefficients reported in Table 8.3. The relative time t^* is defined as $t^* = t^n U_\infty / D$, where t^n is the current timestep. A significant correction in the pressure outside the boundary is observed when the PCA is applied to the expanded IBM with $N_g = 3$, Fig. 8.9a.

Finally, Fig. 8.10 shows the average velocity and velocity gradient profiles at the



(a) **(b)**

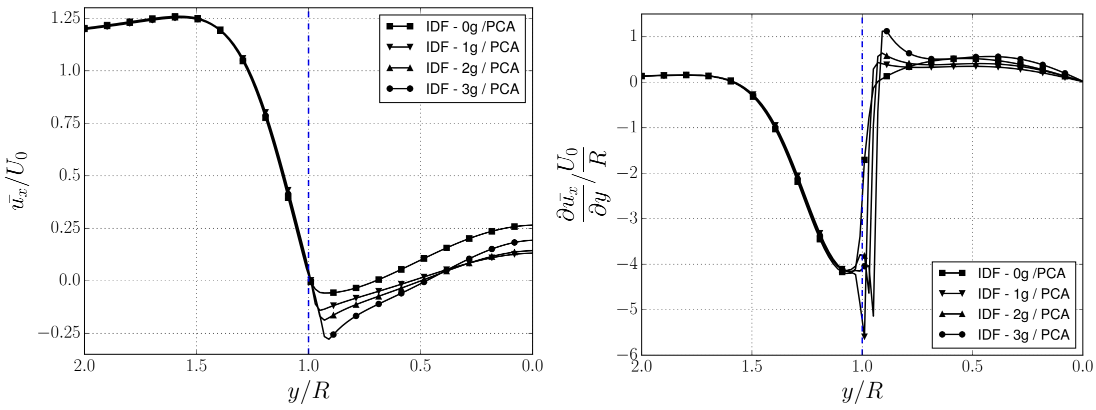
Figure 8.8: (a) Average pressure coefficient and (b) root mean square pressure coefficient on the surface of the cylinder at $Re = 100$ with PCA.



(a) **(b)**

Figure 8.9: (a) Average pressure coefficient at the cross section $x = x_c$ and (b) time evolution of the lift and drag coefficient at $Re = 100$.

cross section of the cylinder at $x = x_c$. The most significant improvements in the velocity gradient profile expansion are noticed at $N_g = 2$ and $N_g = 3$.



(a) (b)

Figure 8.10: (a) Velocity profile and (b) velocity gradient profile at the cross section $x = x_c$ at $Re = 100$.

8.3 Oscillating circular cylinder in a stationary fluid

In this section, the flow around an oscillating circular cylinder in a stationary fluid is considered. The experimental and numerical results of Dütch *et al.* [3] will be used for comparison. The computational domain is $100D \times 100D$, where $D = 100lu$ is the diameter of the cylinder and $lu = 1.0$ is the lattice unit. Periodic boundary conditions have been used for the domain boundaries. The cylinder is placed at the center of the domain and the prescribed oscillating motion is described as in Eq. (7.1). The governing parameters of the flow are the Reynolds number $Re = U_{max}D/\nu$ and the Keulegan-Carpenter number $KC = U_{max}/fD$. The amplitude of the velocity, U_{max}, is derived from Eq. (7.1) as $U_{max} = 2\pi f A_m$. Thus the Keulegan-Carpenter number can be expressed as $KC = 2\pi A_m/D$. We investigate the flow at $Re = 100$ and $KC = 5$. $N_{iter} = 10$ iterations are used in the expanded IBM algorithm.

Fig. 8.11 compares the velocity profile at four cross sections ($x = x_1 - 0.6D, x_1 +$

Table 8.4: Comparison of drag and added-mass coefficients, C_d and C_i, for a translationally oscillating circular cylinder at $Re = 100$ and $KC = 5$.

References	C_d	C_i
Dütch *et al.* [3]	2.09	1.45
Uzunoğlu *et al.* [145]	2.10	1.45
Falagkaris *et al.* [56]	2.09	1.45
Present - $N_g = 0$, PCA	2.09	1.45
Present - $N_g = 1$, PCA	2.09	1.45
Present - $N_g = 2$, PCA	2.09	1.45
Present - $N_g = 3$, PCA	2.09	1.45

$0.0D, x_1{+}0.6D, x_1{+}1.2D$) with the experimental results of Dütch *et al.* [3]. Three phase angles ($\phi = 180^o, 210^o, 330^o$) are examined and three ghost nodes $N_g = 3$ are used in the present case. Good agreement between the present results and the experimental data is observed. The symmetric and periodic vortex shedding, as well as the two-dimensional behaviour of the flow for the present parameter set up ($Re = 100$ and $KC = 5$) are well captured by the present numerical scheme.

The computed motion-averaged drag and added-mass coefficients, C_d and C_i are compared with data from the literature in Table 8.4 and very good agreement is observed. The Lagrangian points approximation is used for the computation of the internal forces as shown in Falagkaris *et al.* [56]. Fig. 8.12a shows the computed drag coefficient for one period of oscillation and good quantitative agreement with the computational results of Dütch *et al.* [3] is observed. In Fig. 8.12b, the pressure correction at $x = x_1 + 0.0D$ and $\phi = 0.75$ is presented.

Finally, the effect of the pressure correction algorithm on the computed pressure coefficient on the surface of the cylinder at four phase angles is investigated. Fig. 8.13 shows the computed instantaneous vorticity at $\phi = 0.625, 0.75, 0.875, 1.0$ using three ghost nodes $N_g = 3$. Respectively, the computed surface pressure coefficient at $\phi = 0.625, 0.75, 0.875, 1.0$ is presented in Fig. 8.14 for $N_g = 1, 3$

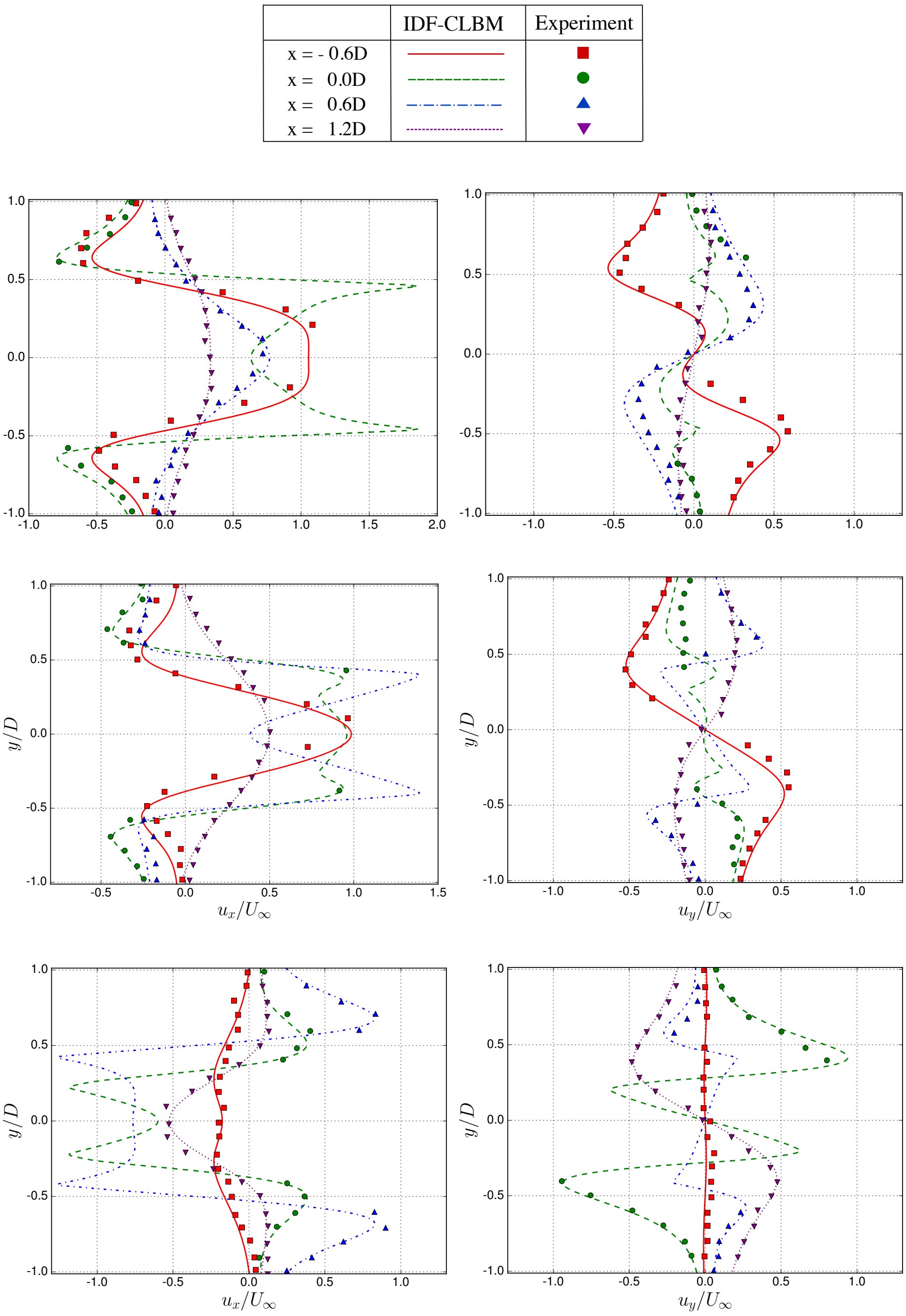
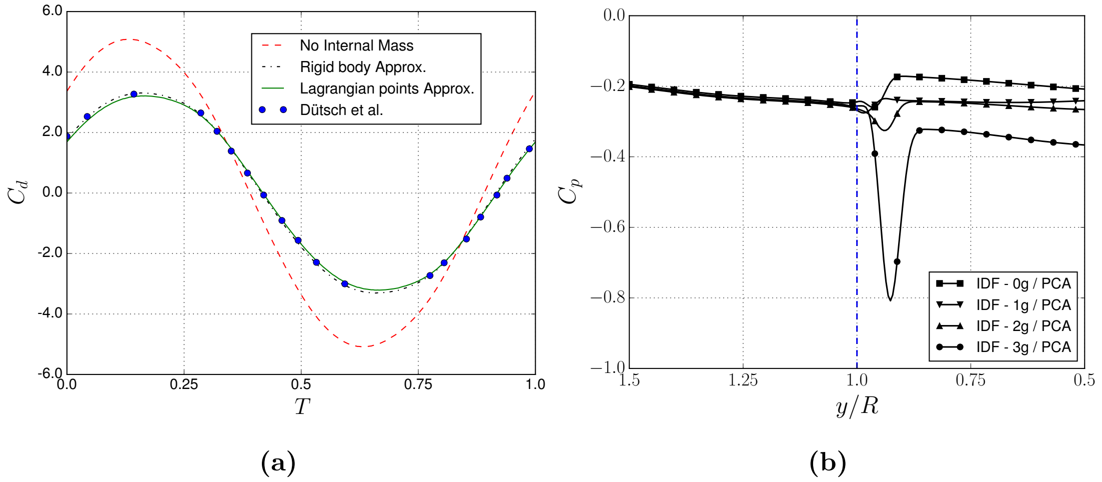


Figure 8.11: Velocity profiles for three phase angles (a) $\phi = 0.5$, (b) $\phi = 0.58$ and (c) $\phi = 0.92$ for $Re = 100$ and $KC = 5$ using three ghost nodes $N_g = 3$ and the PCA scheme. The experimental results of Dütch *et al.* [3] (symbols) are used for comparison.



(a) **(b)**

Figure 8.12: (a) Drag coefficient over a period of a translationally oscillating circular cylinder and (b) Pressure coefficient at the cross section of the cylinder at $\phi = 0.75$ at $Re = 100$ and $KC = 5$ using three ghost nodes $N_g = 3$ and the PCA scheme.

with and without the PCA. The pressure coefficient is computed using Eq. (8.6) by substituting U_∞ with U_{max}. The results verify the observations made in the stationary cylinder case, i.e. accurate pressure values are obtained when the PCA is used under any ghost node configuration at moving boundary applications. Significant corrections in the surface pressure are obtained, especially in the second case $N_g = 3$. It should be noted that the present PCA scheme does not introduce any spurious oscillations near the immersed surface.

8.4 Oscillating circular cylinder in a cross flow

In this section, the flow around a transversely oscillating circular cylinder to an incident stream is examined. The results from the flow around a stationary cylinder at $Re = 100$ are used in order to define the frequency ratio of the oscillations. In this study, only forced oscillations will be examined, i.e. the

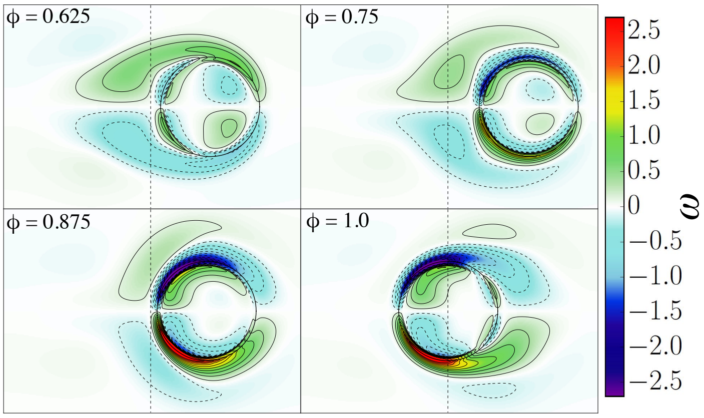
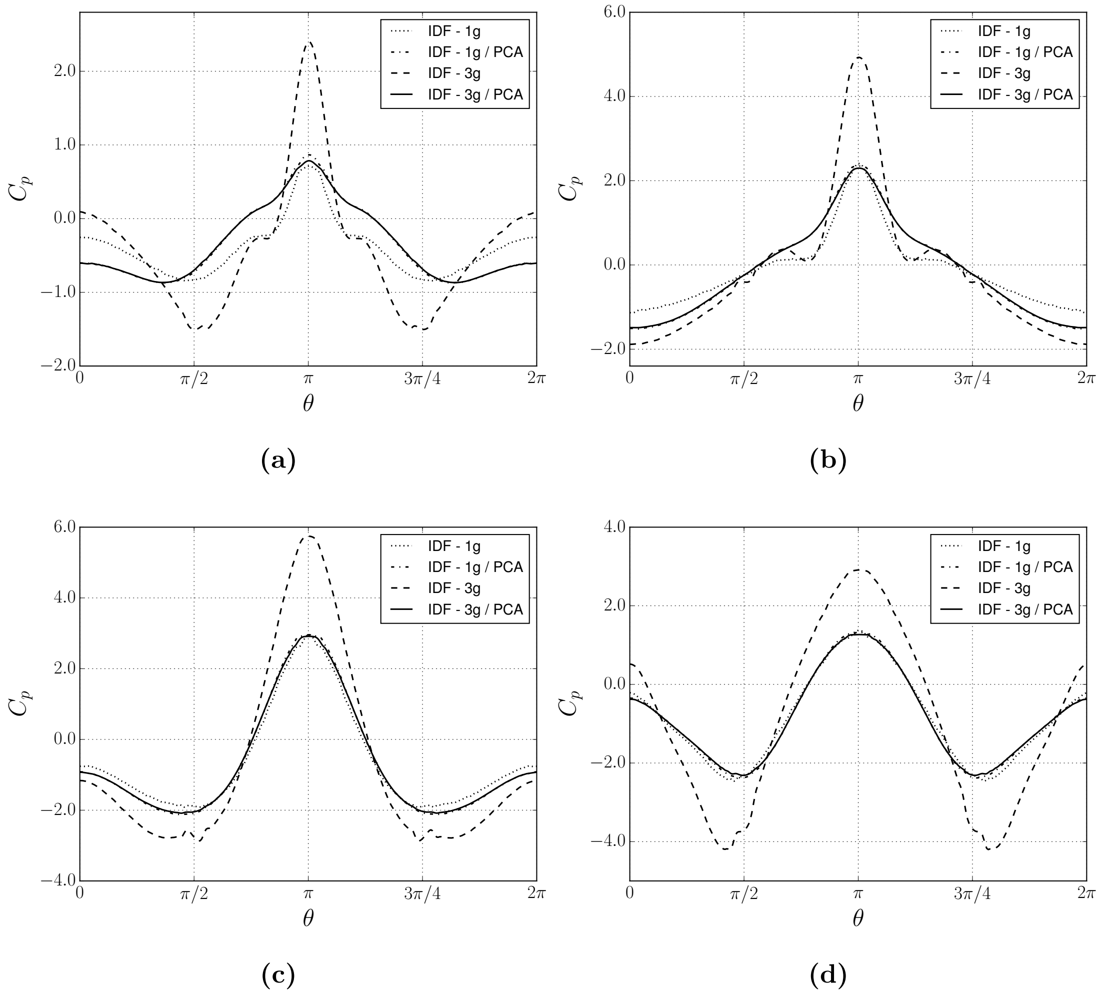


Figure 8.13: Instantaneous vorticity for four phase angles $\phi = 0.625$, $\phi = 0.75$, $\phi = 0.875$ and $\phi = 1.0$.

motion of the cylinder is prescribed as

$$Y_c(t) = A_m sin(2\pi f t), \tag{8.8}$$

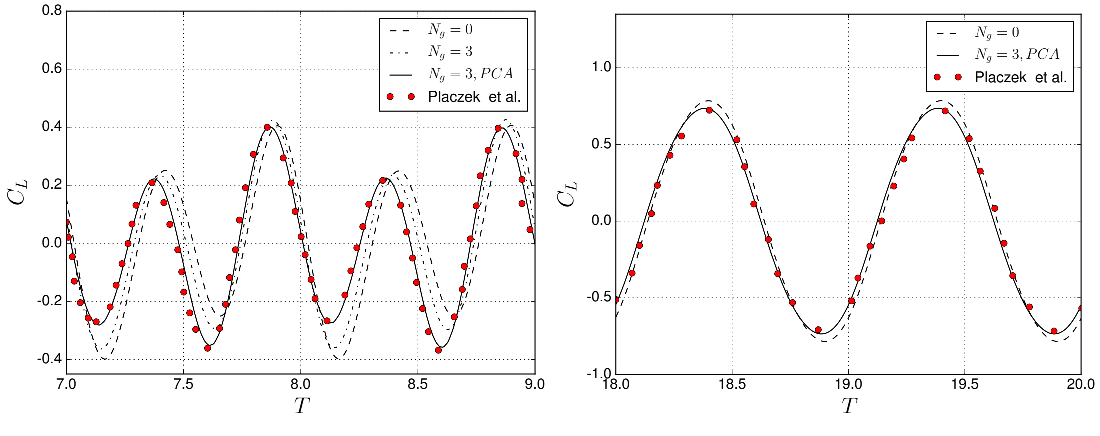
where $A_m = A_0 D$ is the maximum imposed vertical displacement and A_0 is the maximum adimensional amplitude of the oscillation. The governing parameters of the flow are the Reynolds number $Re = U_{max}D/\nu$ and the frequency ratio of the oscillation $F = f/f_s$. We investigate the flow at $Re = 100$ where the Strouhal frequency $f_s = 0.164$ is computed from the stationary cylinder case at the same Reynolds number. The amplitude of the oscillation in set at $A_0 = 2.5$. The vortex structure in the wake regimes are characterized by the amplitude of the oscillation A_0 and the frequency ratio of the oscillation F. According to Koopmann [159] a lock-in zone is defined by the region where the frequency of the vortex shedding diverges from the expected values f_s at the Reynolds number considered and locks on the frequency of the forced oscillations f. For the present configuration at $Re = 100$ and $A_0 = 2.5$ the lock-in zone is limited approximately between $0.75 \leq F \leq 1.25$ [159] and is characterised by a purely sinusoidal evolution of the aerodynamic coeeficients. Two frequency ratios will be examined: (a) $F = 0.5$ out of the lock-in region and (b) $F = 1.1$ in the lock-in



(a) (b) (c) (d)

Figure 8.14: Pressure coefficient on the surface of the cylinder at $\phi = 0.625, 0.75, 0.875, 1.0$ using one and three ghost nodes with PCA.

region. The respective maximum velocities are $U_{max} = 0.0129$ for $F = 0.5$ and $U_{max} = 0.0283$ for $F = 1.1$. The cylinder in placed at the center of a $100D \times 100D$ computational domain. The inlet, outlet and free slip boundary conditions, as described in Chapter 2, have been used for the domain boundaries. The results are compared with the computational results of Placzek *et al.* [160]. The authors reported super-estimated values, compared to analytical expressions and other numerical schemes, for the Strouhal number and the drag coefficient around a



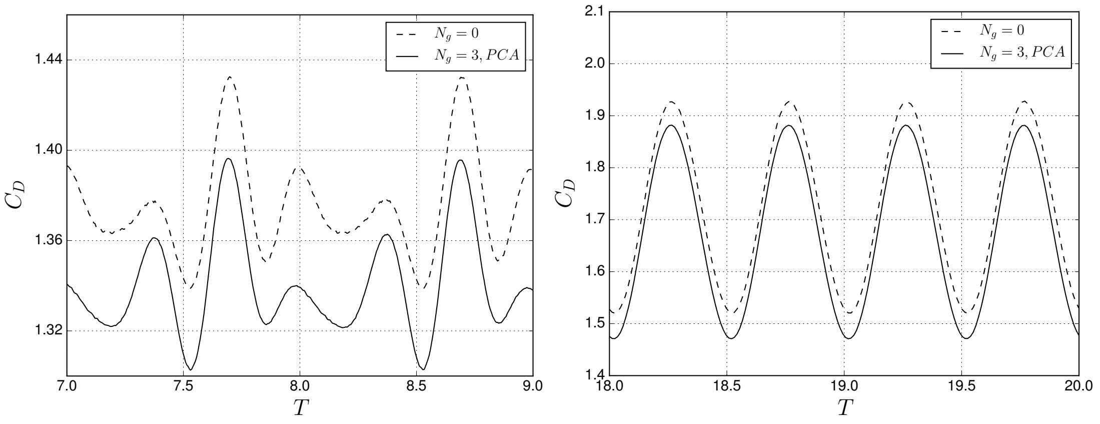
(a) (b)

Figure 8.15: Time evolution of the lift coefficient over two periods of oscillation for (a) $F = 0.5$ and (b) $F = 1.1$ at $Re = 100$.

stationary cylinder at $Re = 100$. The authors related the relative error to the small domain aspect ratio used in their study.

As shown in Fig. 8.15b, the time evolution of the lift coefficient at $F = 1.1$ is characterised by a pure sinusoidal response. Good agreement between the present schemes and the results of Placzek *et al.* [160] is observed. An improvement in the computed lift coefficient is observed when the PCA algorithm is coupled with the IBM. For $N_g = 3$ with the PCA scheme, the maximum value of the lift coefficient increases to $C_L(max) = 0.734$ (compared to $C_L(max) = 0.333$ for the stationary cylinder) and is in good agreement with the reported value $C_L(max) = 0.72$ in [160]. Similar observation can be made for the time evolution of the drag coefficient in Fig. 8.16b. Table 8.5 summarises the average computed drag and maximum lift coefficients. As expected, lower average drag coefficient values are obtained with the present scheme compared to [160].

Fig. 8.15a and Fig. 8.16a show the time evolution of the lift and drag coefficients respectively at the frequency ratio $F = 0.5$. In this case, the signals are not periodic. A beating behaviour, where the signal is not periodic over two successive



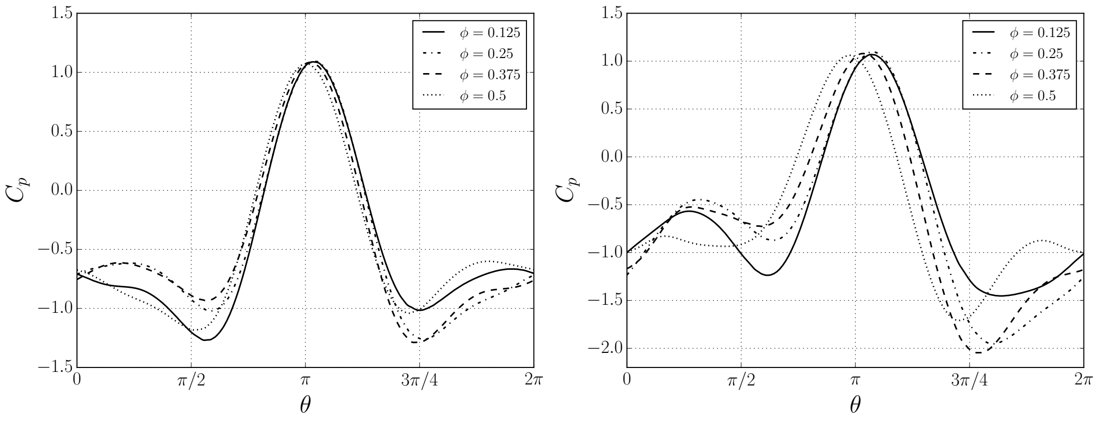
(a) **(b)**

Figure 8.16: Time evolution of the drag coefficient over two periods of oscillation for (a) $F = 0.5$ and (b) $F = 1.1$ at $Re = 100$.

Table 8.5: Comparison of maximum lift and average drag coefficients for the flow over an oscillating circular cylinder in a crossflow at $Re = 100$.

Case	Coefficient	Ref	$N_g = 0, PCA$	$N_g = 1, PCA$	$N_g = 3, PCA$
$F = 0.5$	$\bar{C_D}$	1.379	1.368	1.337	1.345
	$C_L(max)$	0.411	0.407	0.409	0.404
$F = 1.1$	$\bar{C_D}$	1.750	1.702	1.688	1.682
	$C_L(max)$	0.720	0.760	0.744	0.734

oscillations is observed. Similar observations have been made by Anagnostopoulos [161] and Placzek *et al.* [160] for frequencies outside the lock-in zone leading to a quasi-periodic flow pattern over subsequent cycles of oscillations. As shown in Fig. 8.15a the computed lift coefficient using $N_g = 3$ and the pressure correction algorithm are in good agreement with the results obtained by Placzek *et al.* [160] over the oscillating periods $T = 7.0$ to $T = 9.0$. The lift coefficient profiles for $N_g = 0$ and $N_g = 3$ without the PCA scheme, where the minimum values succeed the maximum values after $t = 0.25T$ have been identified by Placzek *et al.* [160] at previous oscillating periods.



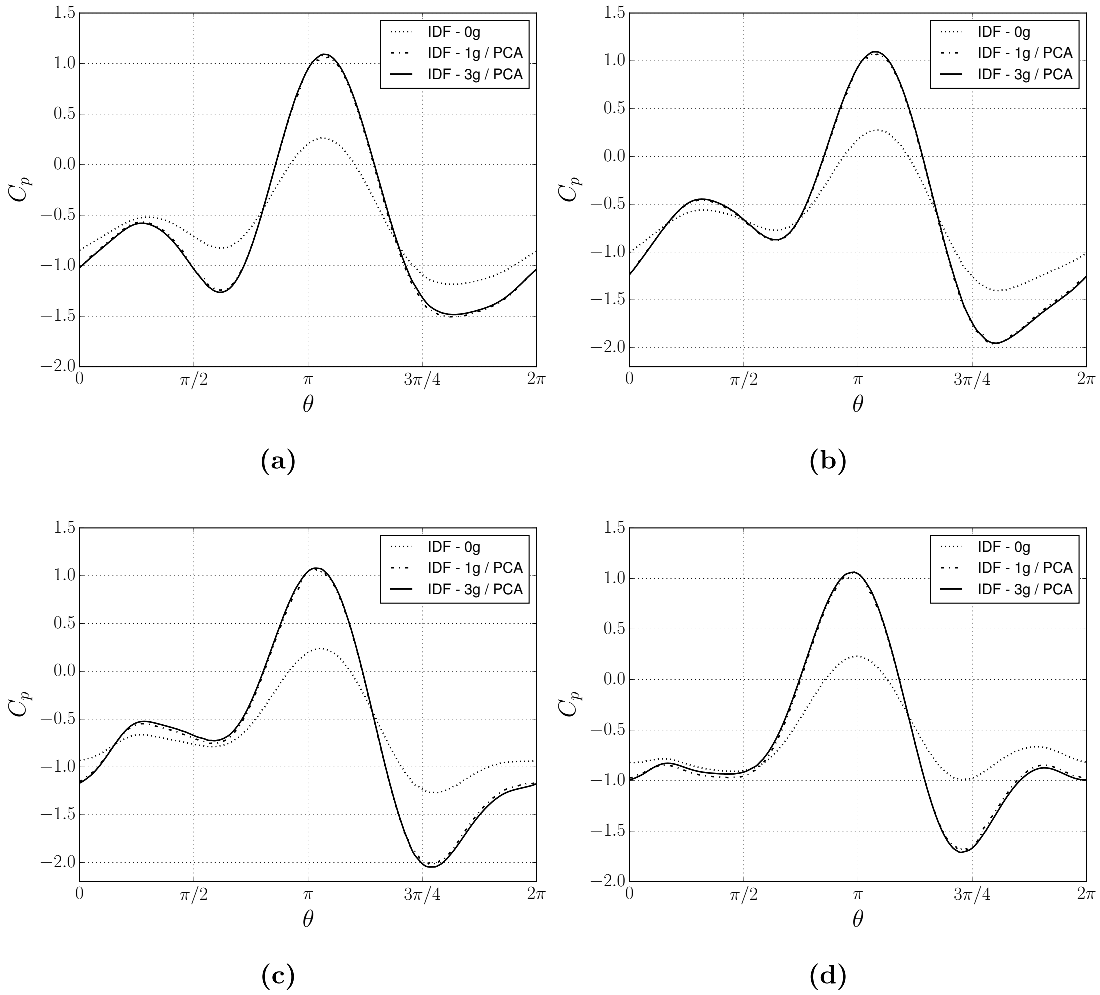
(a) **(b)**

Figure 8.17: Surface pressure coefficient at phase angles $\phi = 0.125, 0.25, 0.375, 0.5$ for (a) $F = 0.5$ and (b) $F = 1.1$ at $Re = 100$.

The effect of the PCA on the computed surface pressure coefficient at $F = 1.1$ is presented in Fig. 8.18 for phase angles $\phi = 0.125, 0.25, 0.375, 0.5$. Significant corrections are observed and good agreement between the two configurations is achieved. Fig. 8.17 shows the computed pressure for both examined frequencies $F = 0.5$ and $F = 1.1$. As expected from the lift coefficient signals, stronger variations and significantly lower pressure values are observed for $F = 1.1$ over a half oscillating period. Finally, Fig. 8.19 shows a sequence of frames for the instantaneous vorticity around the oscillating cylinder during a half oscillation for $F = 0.5$ and $F = 1.1$, in order to identify the differences in the vortex structures in the near wake. The vortex structures at $F = 1.1$ justify the higher lift and drag coefficient values shown in Figures 8.15b and 8.16b.

8.5 Hovering flight of elliptical wings

In this section, the effect of the pressure correction algorithm on the computation of the lift coefficient of an elliptical wing with an aspect ratio of 10 is examined. The elliptical wing undergoes a prescribed sinusoidal rotational and translational



(a) **(b)**

(c) **(d)**

Figure 8.18: Surface pressure coefficient at phase angles $\phi = 0.125, 0.25, 0.375, 0.5$ for $F = 1.1$ at $Re = 100$.

motion in a stationary fluid, prescribed as

$$X(t) = \frac{1}{2}A_m cos(2\pi ft), \quad Y(t) = 0, \tag{8.9}$$

$$\alpha_t = \alpha_0 + \beta sin(2\pi ft + \Phi). \tag{8.10}$$

Eqns. (8.9) - (8.10) represent insect hovering motions. This test case is selected because of the strong dependence of the aerodynamic forces on the pressure [162, 163]. The maximum amplitude of the oscillating motion is set at $A_m = 2.8c$,

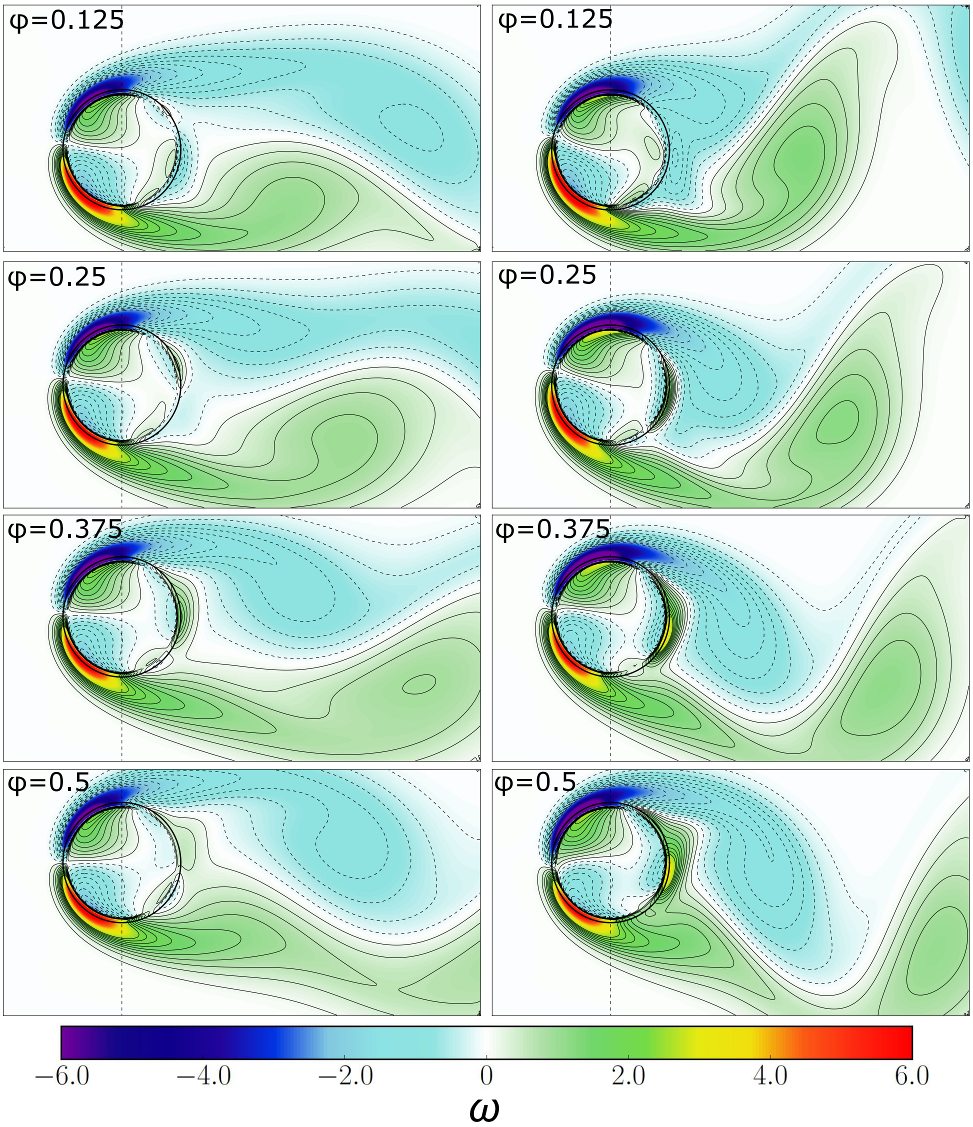


Figure 8.19: Instantaneous vorticity around the oscillating cylinder during a half oscillation at phase angles $\phi = 0.125$, $\phi = 0.25$, $\phi = 0.375$ and $\phi = 0.5$ for $F = 0.5$ (left) and $F = 1.1$ (right).

where c is the chord of the ellipse set as $c = 400lu$. The initial angle α_0 is $\pi/2$ and the rotational amplitude β is $\pi/4$, similar to the simulations and experiments of Wang *et al.* [163] and the simulations of Eldredge [162]. Accordingly, the maximum translational velocity is defined as $U_{max} = \pi A_m/f$, where f is the flapping frequency and $T = 1/f$ is the flapping period. The chord based Reynolds

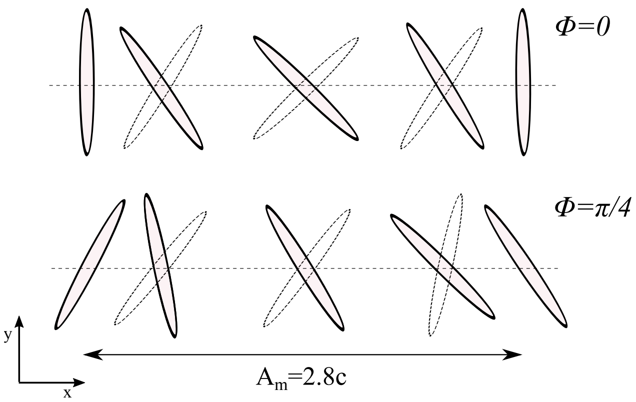
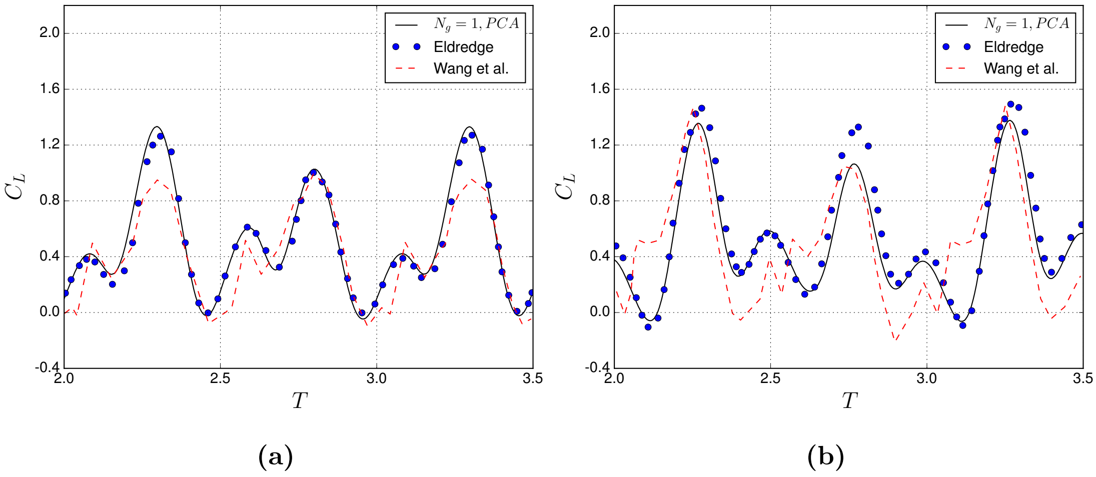


Figure 8.20: Schematic view of the elliptical wing and the prescribed rotational and translational motion.

number is set at $Re = 75$. The elliptical wing in placed at the center of a $100c \times 100c$ computational domain. The inlet, outlet and free slip boundary conditions, as described in Chapter 2, have been used for the domain boundaries.

The phase Φ may be considered as a lag phase between rotation and translation. According to Wang *et al.* [163] the lift force developed by flapping wing motions in hovering kinematics strongly depends on the phase angle Φ. In addition, the unsteady forces, which are strongly dominated by the pressure contribution are expected to reach an almost periodic state after some initial flapping periods. Two phases $\Phi = 0$ and $\Phi = \pi/4$ are investigated. The difference in the motion for $\Phi = 0$ and $\Phi = \pi/4$ is shown in Fig. 8.20. Similar to the previous cases, $N_{iter} = 10$ iterations are used in the expanded IBM algorithm.

Fig. 8.21 shows the time evolution of the lift coefficient over 1.5 periods of oscillation. The expanded IDF-CLBM with $N_g = 1$ and coupled with the PCA is used. The results are compared with the numerical results of Eldredge [162] and the experimental data of Wang *et al.* [163]. Very good agreement between the present scheme and the numerical method of Eldredge [162] is observed. For $\Phi = 0$ in Fig. 8.21a, the computed lift is in overall good agreement with the



(a)

(b)

Figure 8.21: Time evolution of the lift coefficient for (a) $\Phi = 0$ and (b) $\Phi = \pi/4$.

experimental values of Wang *et al.* [163]. However, for $\Phi = \pi/4$ in Fig. 8.21b the lift behaviour between the peaks differs from the experiments [163]. Similar observations have been made by Wang *et al.* [163] between the computed and experimental lift values.

Figures 8.22 and 8.23 show the instantaneous vorticity fields at different phase angles around the ellipse for the symmetrical rotation with $\Phi = 0$ and the advanced rotation with $\Phi = \pi/4$ respectively. The vorticity dynamics for both angles are quite similar, where the same behavior of leading edge vortex recapture and trailing edge vortex shedding is observed [162]. The present wake structure is in good agreement with the vortex shedding patterns in [162].

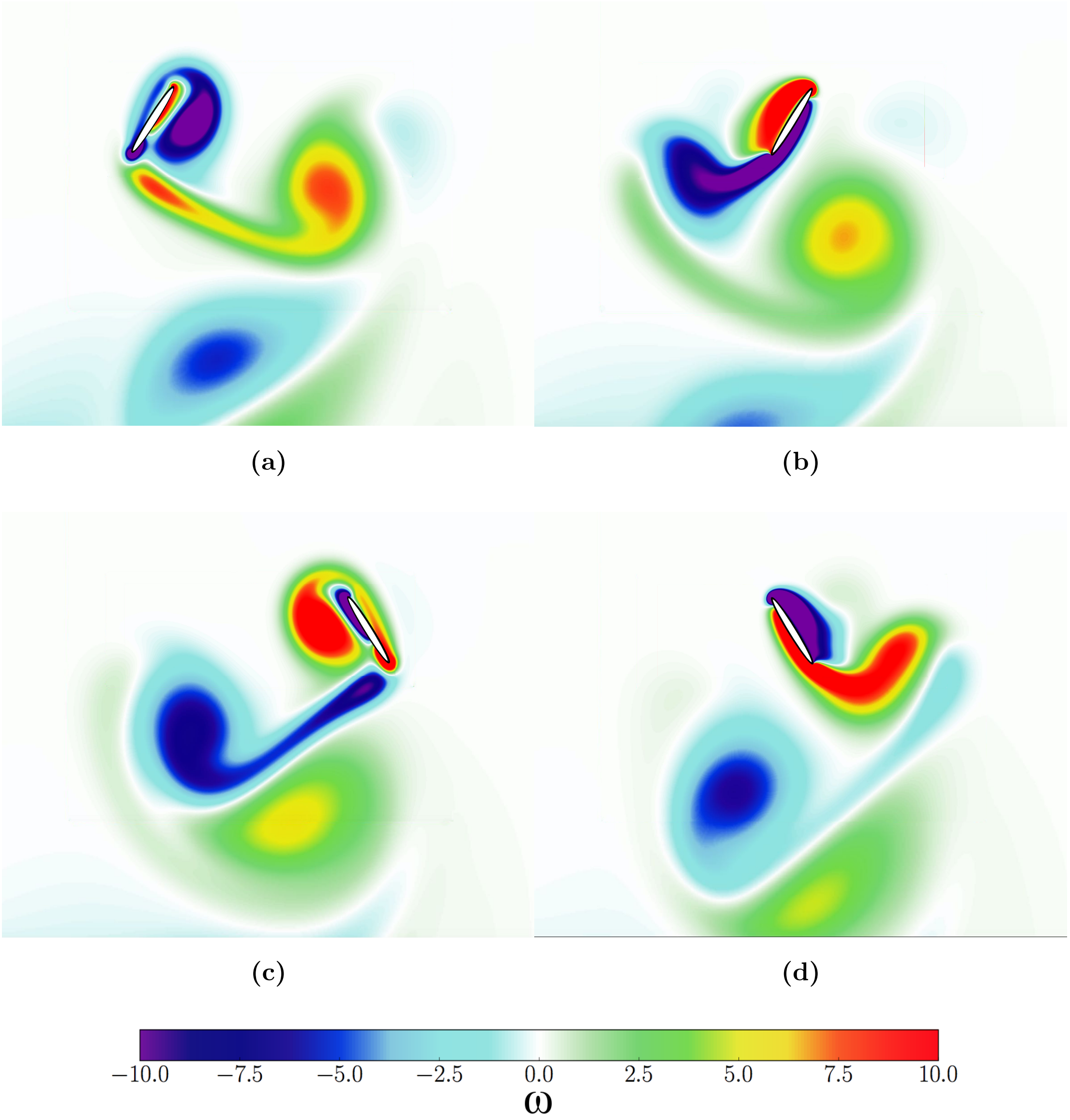
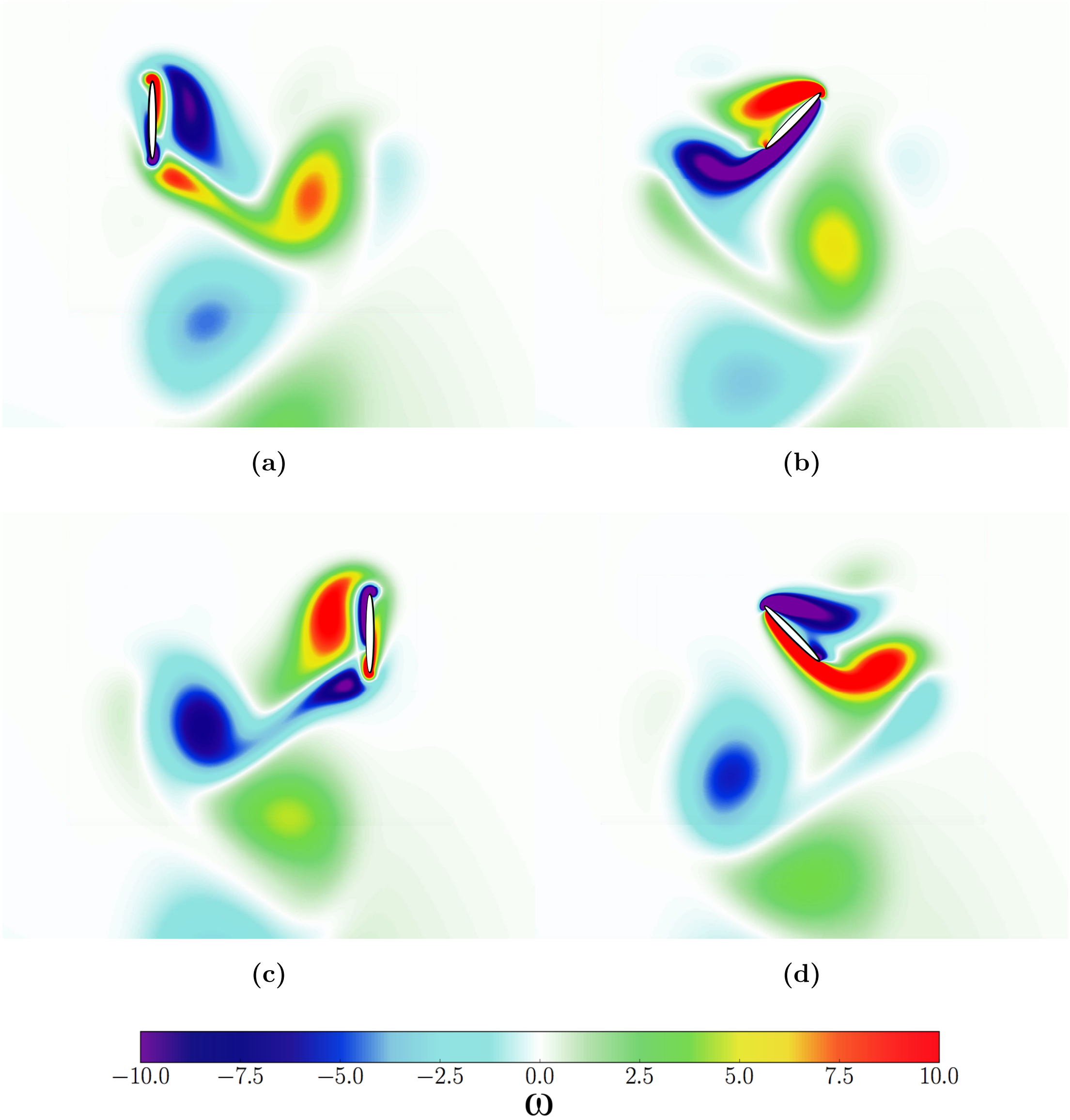

(a) (b) (c) (d)

Figure 8.22: Instantaneous vorticity around the oscillating ellipse during one oscillation at a lag phase $\Phi = \pi/4$ and phase angles (a) $\phi = 0.25$, (b) $\phi = 0.5$, (c) $\phi = 0.75$ and (d) $\phi = 1.0$.



(a) **(b)**

(c) **(d)**

Figure 8.23: Instantaneous vorticity around the oscillating ellipse during one oscillation at a lag phase $\Phi = 0$ and phase angles (a) $\phi = 0.25$, (b) $\phi = 0.5$, (c) $\phi = 0.75$ and (d) $\phi = 1.0$.

Chapter 9

Conclusions and future work

9.1 Conclusions

In the present thesis, a highly accurate and efficient 2-Dimensional numerical coupling between the cascaded multi-domain lattice Boltzmann method and an iterative direct forcing immersed boundary method has been developed. Investigation of flows around stationary and moving boundaries at various Reynolds numbers demonstrated the robustness and stability of the proposed scheme and produced valuable and insightful understanding of the complex flow characteristics and aerodynamic loads acting on 2 dimensional bodies. High resolution accuracy around the investigated flow regions, including efficient resolution of the vortex structures in the wake, has been achieved using a multi-domain algorithm.

To ensure stability and numerical accuracy, the central moment formulation of the LBM [46] [47] has been chosen to solve the fluid dynamics equations. The iterative force correction IBM recently proposed by Zhang *et al.* [103], has been coupled with the CLBM. The numerical accuracy of the boundary treatment has been enhanced by incorporating the effects of both the current and next time step in the discrete external forcing term. Additional improvements to the IDF, including suppression of the velocity gradient discontinuities across the boundaries and corrections on the pressure distribution, have been successfully implemented.

It has been demonstrated that the resulting solver accurately predicts the aerodynamic behaviour of different airfoil designs and allows for very accurate estimations of the changes in the aerodynamic behaviour of the foil by imposing prescribed deformations. The present thesis has been divided into three main sections.

9.1.1 Viscous flows around stationary boundaries

First, viscous flows around stationary boundaries up to high Reynolds numbers have been examined. The robustness and numerical accuracy of the proposed scheme is demonstrated by studies of steady and unsteady flows around a circular cylinder and a NACA-0012 airfoil over a range of Reynolds numbers. The proposed IDF scheme is also compared with an established multi direct forcing IBM [79]. The study of the flow around a circular cylinder indicates that both the IDF and the MDF schemes are in good agreement with other numerical and experimental results in the literature. At Reynolds number of 100 and 150 (the limits for 2D flow), the results obtained with the present method are in better agreement with the results reported by Liu *et al.* [139], using a body-fitted NSE solver, than those from other IBM schemes presented in the literature. Four test cases are investigated for the flow around a NACA-0012 airfoil. At a low Reynolds of 500, the computed pressure coefficient and the boundary layer velocity profiles are in very good agreement with other numerical methods. The computed boundary layer thickness, Fig. 6.11, is identical to the analytical Blasius solution for a flat plate up to the point of maximum curvature on the airfoil surface. Beyond that point, as expected, the pressure gradient adversely affects the thickness of the boundary layer. The aerodynamic coefficients for the moderate Reynolds number cases (1000 and 5000) are accurately captured by the present scheme. The robustness of our scheme is demonstrated using a Reynolds number of 100000. A statistical analysis of the time dependent solution indicates that the IDF-CLBM accurately reproduces the unsteady lift and drag behaviour reported by Mittal and Tezduyar [140].

9.1.2 Viscous flows around moving and wilfully deformable boundaries

Second, the present solver is applied to viscous flows around moving and wilfully deformed boundaries. We have extensively investigated the effects of the internal mass in the computation of the aerodynamic loads and shown that the Lagrangian points approximation scheme significantly improves the numerical accuracy over a wide range or Reynolds numbers and frequency-amplitude motion configurations. The robustness and numerical accuracy of the present scheme is demonstrated by studies of unsteady flows around an oscillating cylinder, a pitching foil, a plunging foil and a plunging and flapping foil. The present results compare well with other experimental and numerical results found in the literature.

The importance of the internal mass effects on the computation of the aerodynamic loads of moving boundaries, where the boundary motion is prescribed, has been demonstrated. Significant differences both in the magnitude phase of the loads have been found. Overall, a strong dependence of the internal forces on the motion characteristics (amplitude of oscillation and frequency) has been observed. In linear motions, both the rigid body (RBA) and the Lagrangian points (LPA) approximations accurately estimate the linear momentum of the immersed body. For the flow around an oscillating cylinder, at $Re = 100$, and around a plunging SD7003 foil, at $Re_c = 40000$, the aerodynamic loads have been accurately computed with both schemes. Neglecting the internal mass effects results in over-prediction of the aerodynamic loads by $10\% - 50\%$. However, as the internal velocity is strongly influenced by the force spreading operation in the IBM, an approximation of the internal momentum with a prescribed value would lead to inaccuracies in more complex motions. The present IDF-CLBM solver has been shown to accurately capture the vortex dynamics around the plunging SD7003 foil and good agreement between the computed and the experimental time-averaged wake profiles has been observed. In angular and wilfully deformed motions, where the rigid body approximation is not valid, the momentum of the internal fluid has been successfully computed with the Lagrangian point approximation scheme. Initially, the flow around a pitching foil has been examined and compared with experiments. The computed drag coefficient LPA values have been found to be in good agreement with experimental values. The vortex structure computed with the present IDF-CLBM scheme agrees well with the experimental structure for a wide range of flapping amplitudes. Finally, the flow around a flapping wilfully deformed NACA-0014 airfoil, at $Re = 10000$, and various amplitude/deformation configurations has been investigated in order to demonstrate the robustness of our scheme. As expected, a strong dependence of the aerodynamic loads on the leading and trailing-edge vortex dynamics has been observed. Overall, the periodic profile of the aerodynamic loads is strongly related to the motion characteristics and is significantly affected by the internal mass effects.

The LPA is shown to be an accurate and flexible method of incorporating the effects of the internal mass in the computation of the aerodynamic loads around moving objects. We conclude that for flows around moving and wilfully deformed bodies, where the motion of the body is prescribed, using 25% of the total internal Lagrangian points is sufficient to capture the momentum of the internal fluid whilst maintaining the accuracy of the computation and reducing the computational overhead by 4%.

9.1.3 Higher-order immersed boundary method with pressure correction

Finally, a hybrid immersed boundary method has been developed in order to address two important drawbacks of the IBM. An expanded velocity profile scheme based on the method of Suzuki and Inamuro [130] has been derived in order to compensate for the discontinuities caused by the velocity gradient across the boundary. Second, a pressure correction algorithm has been presented in order to accurately prescribe Neumann pressure boundary conditions in the immersed boundary framework (presence of external forces) and correct the pressure distribution across the boundary.

First, the flow around a stationary circular cylinder at $Re = 40$ (steady flow) and $Re = 100$ (unsteady flow) has been examined. It has been found that in order to achieve the prescribed boundary conditions (no slip), less iterations in the IBM algorithm are required when the expanded scheme is used. Four configurations have been investigated using $N_g = 0, 1, 2, 3$ ghost nodes inside the boundary. The computed aerodynamic coefficients are in good agreement with experimental and other numerical results for all four configurations. It has been shown that the pressure along the boundary is indirectly affected by the number of ghost nodes used in the expanded IBM. For $N_g = 0$, conventional IBM, the pressure is not expanded correctly across the boundary leading to underestimated pressure coefficient values along the surface of the body. As the number of ghost nodes increases, the pressure distribution across the boundary is somehow arbitrarily improved. However, for $N_g = 3$, over-predicted values have been obtained. When the pressure correction algorithm is applied, the correct pressure values have been recovered for all four configurations. Similar observations can be made for the computed root mean square pressure coefficient at the unsteady flow case at $Re = 100$.

Second, the effect of the pressure correction algorithm on the computation of the aerodynamic coefficients is examined by investigating the flow around an oscillating cylinder in a stationary fluid and in a cross flow. In the first case, the computed results are compared with the experimental and numerical results of Dütch *et al.* [3] and very good agreement has been observed. As shown, the pressure correction algorithm effectively corrects the pressure distribution across the boundary. It should be noted that compared to the scheme of Tiwari and Vanka [131] our method does not introduce any spurious pressure oscillations near the immersed surface. Some interesting observations can be made for the flow around an oscillating cylinder in a cross flow. Two frequency ratios $F = 0.5$ (unlocked configuration) and $F = 1.1$ (locked configuration) have been examined showing an overall good agreement between the present scheme and the computational results of Placzek *et al.* [160]. For the unlocked configuration case at $F = 0.5$, a significant improvement in the computed lift coefficient is observed when the pressure correction algorithm is coupled with the expanded IBM.

Finally, the flow around an elliptical wing undergoing a prescribed sinusoidal rotational and translational motion has been examined. According to Wang *et al.* [163] and Eldredge [162] the aerodynamic forces strongly depend on the pressure distribution around the boundary. We investigated the effect of the pressure correction algorithm on the computed lift coefficient and good agreement with the computational results of Eldredge [162] and the experimental data of Wang *et al.* [163] has been found.

Significant improvements in the computation of the aerodynamic coefficients at moving boundary applications is achieved. The proposed method of smoothly reconstructing the boundary layer profile inside the solid domain effectively suppresses the velocity gradient discontinuities away from the boundary and increases the accuracy of the solution. Investigation of the flow around stationary and moving boundaries demonstrate the drastic improvements on the pressure distribution across the boundary with the proposed scheme. The proposed PCA algorithm is very flexible and can be implemented to any IBM. Application of the proposed hybrid scheme to higher Reynolds number flows and fully coupled fluid structure interaction problems would be of great interest.

9.2 Future work

This thesis has been mainly focused on developing a novel, robust and accurate 2D numerical code for the simulation of viscous flows around moving and wilfully deformed boundaries up to moderate/high Reynolds numbers. Future work concerns expansion of the present scheme to 3 dimensions and incorporation of additional physics. That includes

1. **Expansion to 3 dimensions**. The recently developed D3Q27 cumulant LBM of Geier *et al.* [48] may be considered as an improvement of the central moment formulation LBM. According to Geier *et al.* [48] the cumulant LBM has been shown to have extremely good stability and adaptivity properties at high Reynolds number flows, while maintaining or even improving the computational accuracy of the cascaded LBM. The expansion to 3D will require further development of the multi-domain algorithm with more advanced grid coupling techniques, e.g. the bubble functions for the interpolation of the velocity field, see Geier *et al.* [109]. In addition, more accurate geometrical representations of the 3 dimensional bodies will be considered.

2. **Efficient parallel programming**. This will require further research

into parallel programming architectures in order to optimise the numerical efficiency of the 3D numerical code. Initial considerations based on the local nature of the LBM include implementation on Graphics Processing Units (GPUs).

3. **Fluid structure interaction**. Additional physics should be implemented including a Finite Element Method (FEM) solver for the solution of the structural dynamics and an surface energy minimisation algorithm for the coupling between the fluid and the solid solvers.

4. **Additional physics**. Incorporation of additional physics such as *Thermal LBM, Free surface flows* and *Turbulence modelling* should be examined. Potential implementation of a dynamic turbulence model for high Reynolds number flows should be considered.

   Results of the present scheme for high Reynolds number flows around 2 dimensional airfoils indicate the applicability of the scheme to 3 dimensional flows. In particular, the computed aerodynamic coefficients around the oscillating SD7003 airfoil are in very good agreement with the 3D data and the experimental results of Visbal [6]. Regarding the vortex structures in the near wake, the good agreement between the present results and the experiments of McGowan *et al.* [5] indicate the ability of the cascaded LBM scheme to accurately resolve 2 dimensional vortex structures at high Reynolds numbers. However, based on the current results, direct 3D predictions can not be made without further testing and expanding of the code to the cumulant 3 dimensional LBM. The CLBM may be considered as a MRT-LBM where collisions are performed in the space of central moments. As shown in Geier *et al.* [48] the accuracy and robustness of the raw moment MRT scheme significantly deteriorates if the relaxation rates are not optimally specified. This issue is eliminated with the central moment and by extension the cumulant LBM formulations, where all relaxation parameters, except the one connected with the shear viscosity, can be set to unity without compromising the numerical accuracy. Finally, the selection of the relaxation rates to equal unity is not determined by requirements to delete errors from Galilean invariance or hyper-viscosity restrictions. Potentially, the raw moment MRT LBM could produce similarly accurate results for the present investigated cases by carefully testing and individually choosing the relaxation rates for each examined flow case.

The potential need of a turbulence model in order to improve the numerical stability, accuracy and robustness in the 3 dimensional expansion of the proposed scheme should be further investigated. In addition, the effect of a sub-grid-scale model on the immersed boundary method should be carefully considered.

Appendix A

A.1 Scaling in the lattice-Boltzmann method

The Lattice Boltzmann equation is known to recover the Navier-Stokes equations with second order accuracy in the Knudsen number ϵ using a second order accurate quadrature rule and proper coupling of the space and the time step from the discrete Boltzmann equation [171]. Therefore, the compressible Navier-Stokes equations are recovered with a formally second order accurate scheme in time and space. Expressing the overall error in terms of the truncation errors in time and space and the error in the Knudsen number, $E_t = \mathcal{O}(\Delta t^2)$, $E_x = \mathcal{O}(\Delta x^2)$ and $E_\epsilon = \mathcal{O}(\epsilon^2)$ respectively we get

$$E = E_t + E_x + E_\epsilon = \mathcal{O}(\Delta t^2) + \mathcal{O}(\Delta x^2) + \mathcal{O}(\epsilon^2). \tag{A.1}$$

Assigning L as a characteristic length of the flow the Knudsen number in the LBM framework may be interpreted as $\epsilon = \Delta x/L$ [171] and thus $E_\epsilon = \mathcal{O}(\epsilon^2) = \mathcal{O}(\Delta x^2)$. The overall error may now be expressed as

$$E = E_t + E_x = \mathcal{O}(\Delta t^2) + \mathcal{O}(\Delta x^2). \tag{A.2}$$

Before proceeding with the effect of the scaling on the overall error let us first introduce the compressibility errors E_{Ma} occurring from the low Mach number ($Ma < 0.3$) approximation of the LBM schemes presented in the thesis. This error is introduced when incompressible flows are investigated with the LBM due to the fact that LBM schemes are inherently compressible scheme as the density is not constant and is related to the pressure through an equation of state. According to [172, 173] the compressibility error is of order $E_{Ma} = \mathcal{O}\big(\frac{\Delta t^2}{\Delta x^2}\big)$ and the overall error is

$$E = E_t + E_x + E_{Ma} = \mathcal{O}(\Delta t^2) + \mathcal{O}(\Delta x^2) + \mathcal{O}\big(\frac{\Delta t^2}{\Delta x^2}\big). \tag{A.3}$$

Let us now discuss the effect of the diffusive and acoustic scaling on the overall error. In the diffusive scaling the discrete time step is proportional to the square of the discrete space step $\Delta t \propto \Delta x^2$, thus the compressibility error reduces to $E_{Ma} = \mathcal{O}\big(\frac{\Delta t^2}{\Delta x^2}\big) = \mathcal{O}\big(\frac{\Delta x^4}{\Delta x^2}\big) = \mathcal{O}(\Delta x^2)$. The numerical scheme is now formally second order accurate in space. Effectively, when Δx is halved, the Mach number Ma is halved but the speed of sound c_s is doubled. The diffusive scaling does not affect the relaxation rate. However, when incompressible flows are investigated, the numerical scheme under diffusive scaling reduces to only first order accuracy in time [26, 174].

In the convective scaling the discrete time step is proportional to the discrete space step $\Delta t \propto \Delta x$, thus the compressibility error reduces to $E_{Ma} = \mathcal{O}(1)$. The speed of sound c_s and the Mach Ma and Reynolds Re numbers are constant across grids with different resolutions. However, the compressibility error $E_{Ma} = \mathcal{O}(1)$ may in general be considered as non-convergent [171, 174]. According to [171, 174, 172] a quadratic decrease of the overall error with Δx holds until the compressibility error gets dominant. Therefore, in order to preserve the second order convergence of a LBM scheme under convective scaling must be suppressed. According to [175] this is achieved by keeping the relaxation rate above one $\omega > 1$ in all levels.

Holdych *et al.* [174] shows that in the over relaxation region the compressibility error is not dominant. Numerical validation of this effect can be found in [176, 54].

A.2 Unit conversion in Multi - Grid algorithm

Following the convective (acoustic) scaling in Section 2.4.2 it occurs that $\delta t_f/\delta x_f = \delta t_c/\delta x_c =$ constant. A consequence of the convective scaling is that both the velocity and pressure fields are continuous across grids with different resolution levels. However, the viscosity must be rescaled. Let us define the Reynolds number as $Re_r = U_r L_r/\nu_r$, where r stands for either the coarse c or the fine f grid resolution. U_r, L_r and ν_r are the characteristic velocity, length scale and the viscosity on the r grid. Both U_r and L_r may be written in terms of their respective physical quantities U and L as $U_r = U\delta t_r/\delta x_r$ and $L_r = L/\delta x_r$. Following Lagrava *et al.* [54] and imposing an independent of the grid level Reynolds number we get

$$Re_f = Re_c \longleftrightarrow \frac{UL\delta t_c}{\delta x_c^2 \nu_c} = \frac{UL\delta t_f}{\delta x_f^2 \nu_f}. \tag{A.4}$$

Therefore, using the convective scaling, the rescaling of the kinematic viscosity is described as

$$\nu_f = \frac{\delta x_c}{\delta x_f}\nu_c. \tag{A.5}$$

Finally, remembering the relationship between the viscosity and the relaxation rate $\nu = \frac{1}{3}\left(\frac{1}{\omega} - \frac{1}{2}\right)$ we get

$$\omega_f = \frac{2\delta x_f \omega_c}{\delta x_f \omega_c + 2\delta x_c - \delta x_c \omega_c} = \frac{2\omega_c}{4 - \omega_c}. \tag{A.6}$$

In the LBM, the distribution functions can be written as a summation of the equilibrium and non-equilibrium parts as $f_{\alpha,m} = f_\alpha^{eq}(\rho_m, \mathbf{u}_m) + f_{\alpha,m}^{neq}(\nabla \mathbf{u}) = f_\alpha^{eq}(\rho, \mathbf{u}) + f_{\alpha,m}^{neq}(\nabla \mathbf{u})$, where $\mathbf{u}_f = \mathbf{u}_c = \mathbf{u}$ and $\rho_f = \rho_c = \rho$ since both fields are continuous and independent of the grid resolution.

The non-equilibrium part f_α^{neq} is proportional to the velocity gradient. Therefore, in order to ensure continuity of the f_α^{neq} across grids with different resolutions $f_{\alpha,f}^{neq} = \beta f_{\alpha,c}^{neq}$, where β is to be determined, the non-equilibrium parts must be rescaled. From the Chapman Enskog expansion it can be shown that the non-equilibrium $f_\alpha^{neq} = \epsilon f_\alpha^{(1)}$, where $\epsilon \ll 1$, is given by

$$\epsilon f_\alpha(1) = \frac{w_\alpha}{2c_s^4}\mathbf{Q}_\alpha : \mathbf{\Pi}^{(1)}, \tag{A.7}$$

where the tensor $\mathbf{\Pi}^{(1)} \equiv \sum_{\alpha=0}^{q-1} \mathbf{e}_\alpha \mathbf{e}_\alpha \epsilon f_\alpha^{(1)}$ is related to the rate of strain tensor $\mathbf{S}$ as

$$\mathbf{\Pi}^{(1)} = -\frac{2c_s^2\rho}{\omega}\mathbf{S}. \tag{A.8}$$

Finally, the non-equilibrium part is given by

$$f_\alpha^{neq} = -\frac{w_\alpha \rho}{c_s^2 \omega}\mathbf{Q}_\alpha : \mathbf{S}. \tag{A.9}$$

From Eq. (A.9) and using $f_{\alpha,f}^{neq} = \beta f_{\alpha,c}^{neq}$, the factor β may be computed as

$$\beta = \frac{\delta t_c}{\delta t_f}\frac{\omega_c}{\omega_f}. \tag{A.10}$$

Noting that $\delta t_c = 2\delta t_f$ the scaling of the non-equilibrium across grids with different resolution levels may be done as

$$f_{\alpha,f}^{neq} = \frac{\omega_c}{2\omega_f} f_{\alpha,c}^{neq}. \tag{A.11}$$

In the cascaded LBM framework, the rescaling of the non-equilibrium in Eq. (A.11) could be performed in the space of central moments in order to scale only the non-equilibrium moments that relate to the shear and bulk viscosities. A similar approach has been followed by Chen *et al.* [164] for the MRT-LBM. Recently, Kuwata and Suga [165] presented an alternative scaling scheme for the

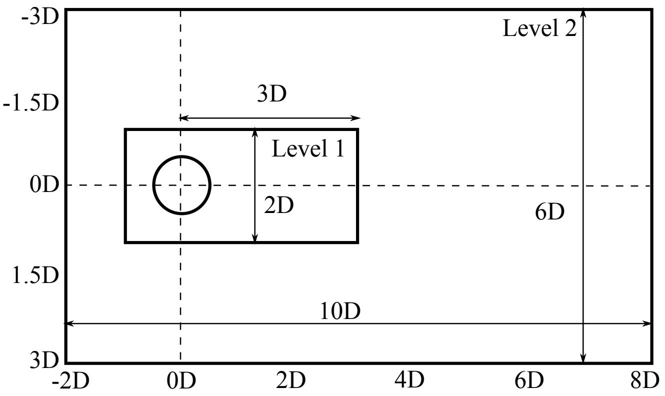


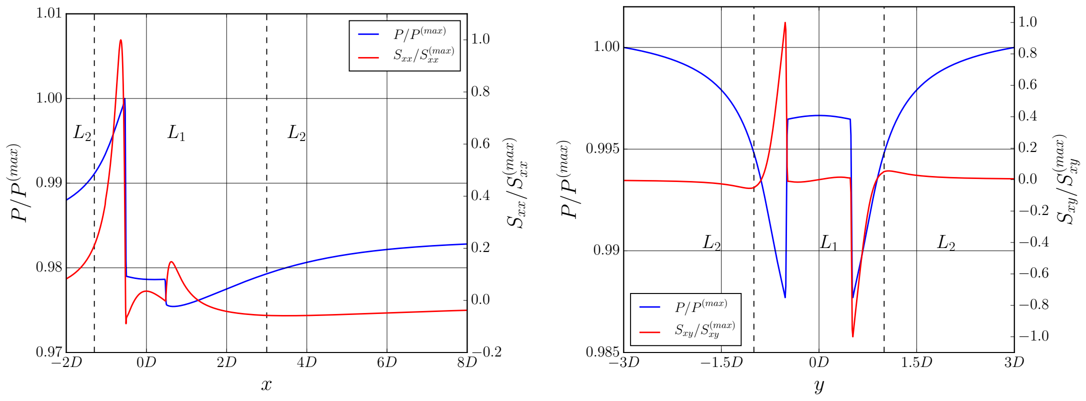
Figure A.1: Grid refinement around a circular cylinder. Two resolution levels are shown.

non-equilibrium parts where a correction step is introduced in order to remove the interface discontinuities of the density and velocity fields.

In order to evaluate the consistency and accuracy of the present numerical scheme let us consider the flow around a stationary cylinder at $Re = 40$ as described in Section 6.2. Fig. A.1 shows the layout of two grid resolution levels around the circular cylinder. Using Eq. (A.6) the relaxation rates for the nine levels of grid refinement are $\omega_{l1} = 0.8$, $\omega_{l2} = 1.1428$, $\omega_{l3} = 1.4545$, $\omega_{l4} = 1.6842$, $\omega_{l5} = 1.8286$, $\omega_{l6} = 1.9104$, $\omega_{l7} = 1.9542$, $\omega_{l8} = 1.9768$ and $\omega_{l9} = 1.9883$. Fig. A.2 shows the pressure and stresses distribution across the grid interface on the x and y directions. The stresses are computed using the non-equilibrium distributions Eq. (A.8). It is observed that both the pressure and the shear stresses are continuous quantities across the interfaces.

A.3 Computational costs

As an indication of the numerical cost of the proposed numerical schemes let us discuss the flow around a stationary cylinder as described in 6.2. The computational domain is $50D \times 50D$, where $D = 100$, with nine levels of grid



(a) **(b)**

Figure A.2: Pressure and Stress distribution across the grid interface on (a) the x direction at $y = y_c$ and (b) the y direction at $x = x_c$ for the flow around a circular cylinder at $Re = 40$.

refinement. Please note that this is not an optimal grid configuration and further improvements could be made in order to reduce the computational cost of the method.

Using the convective grid scaling one time iteration on the ninth level t_{L_9} corresponds to $t_{L_1} = 2^8 = 256$ iterations on the finest grid level. The CPU time of one full time iteration on the $L1$ grid with 405×205 grid points is $t_{L_1} = 0.167$ sec. Let us write $t_{L_1} = t_{L_1}^{(bulk)} + t_{L_1}^{(IBM)} + t_{L_1}^{(L_1/L_2)}$, where $t_{L_1}^{(bulk)}$, $t_{L_1}^{(IBM)}$ and $t_{L_1}^{(L_1/L_2)}$ refer to the bulk flow, the immersed boundary method and the grid communication respectively. For this specific grid configuration $t_{L_1}^{(L_1/L_2)} \approx 12\% t_{L_1}$. The immersed boundary time is $t_{L_1}^{(IBM)} = 0.03$ sec for 312 Lagrangian nodes and $N = 20$ iterations using the IDF scheme. The net bulk flow time for the CLBM is then $t_{L_1}^{(bulk)} = 0.12$ sec. On a L_9 grid time iteration the cost of the bulk flow and the IB algorithm is $t_{L_9}^{(bulk)} = 30.7$ sec and $t_{L_9}^{(IBM/IDF)} = 7.68$ sec respectively. Using the MDF immersed boundary scheme, the computational cost would decrease to $t_{L_9}^{(IBM/MDF)} = 5.23$. For this specific, non-optimal, grid configuration the allocated time on the L_1 grid is approximately $1/3$ of the total simulation time $t_{L_1} = 1/3t^{(Tot)}$. Therefore, the overall numerical efficiency of the numerical scheme could be drastically improved by optimising the grid configuration.

Finally, let us discuss the effect of the proposed higher-order IBM on the overall computational time of the scheme. Using ghost nodes to smoothly expand the velocity profile inside the solid domain increases the immersed boundary time under a single L_1 step as $t^{(IBM)}_{L_1}(g = 0) = 0.03$ sec, $t^{(IBM)}_{L_1}(g = 1) = 0.054$ sec, $t^{(IBM)}_{L_1}(g = 2) = 0.078$ sec, $t^{(IBM)}_{L_1}(g = 3) = 0.102$ sec. The pressure correction algorithm is not an iterative scheme but includes the interpolation and spreading of non-equilibrium distributions, which is the most expensive process in the scheme. The overall cost of the PCA algorithm is $t^{(PCA)}_{L_1} = 0.0123$ sec. Incorporating both the PCA algorithm and the IBM with $N_g = 3$ ghost nodes in the algorithm, the CPU time of one full time iteration on the $L1$ grid increases from $t_{L_1} = 0.167$ sec to $t_{L_1} = 0.254$ sec. The increase in the overall computational is approximately 14%.

Bibliography

[1] U. "Ghia, K.N Ghia, and C.T" Shin. High-re solutions for incompressible flow using the navier-stokes equations and a multigrid method. *J. Comput. Phys.*, 48(3):387 – 411, 1982.

[2] T. Imamura, K. Suzuki, T. Nakamura, and M. Yoshida. Flow Simulation Around an Airfoil by Lattice Boltzmann Method on Generalized Coordinates. *AIAA Journal*, 43(9):1968–1973, September 2005.

[3] H. Dütsch, F. Durst, S. Becker, and H. Lienhart. Low-Reynolds-number flow around an oscillating circular cylinder at low KeuleganCarpenter numbers. *J. Fluid Mech.*, 360:249271, 1998.

[4] R. Godoy-Diana, J.-L. Aider, and J. E. Wesfreid. Transitions in the wake of a flapping foil. *Phys. Rev. E*, 77:016308, Jan 2008.

[5] G. McGowan, A. Gopalarathnam, V. Ol. Michael, J. Edwards, and D. Fredberg. Computation vs. Experiment for High-Frequency Low-Reynolds Number Airfoil Pitch and Plunge. In *46th AIAA Aerospace Sciences Meeting and Exhibit*. AIAA, 2008.

[6] M. Visbal. High-Fidelity simulation of transitional flows past a plunging airfoil. In *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*. AIAA, 2009.

[7] Fung Y. C. *An Introduction to the Theory of Aeroelasticity*. Dover, 1993.

[8] "R. Benzi, S. Succi, and M. Vergassola". The lattice boltzmann equation: theory and applications. *Physics Reports*, 222(3):145 – 197, 1992.

[9] Shiyi Chen and Gary D. Doolen. Lattice boltzmann method for fluid flows. *Annual Review of Fluid Mechanics*, 30(1):329–364, 1998.

[10] S. Succi. *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond.* Clarendon Press, 2001.

[11] D. Yu, R. Mei, L.-S. Luo, and W. Shyy. Viscous flow computations with the method of lattice Boltzmann equation. *Progress in Aerospace Sciences*, 39(5):329–367, July 2003.

[12] C. Cercignani. *The Boltzmann Equation and Its Applications*, volume 67 of *Applied Mathematical Sciences.* Springer, 1988.

[13] D. A. Wolf-Gladrow. *Lattice-Gas Cellular Automata and Lattice Boltzmann Models: An Introduction.* Springer, 2004.

[14] T Reis and T N Phillips. Lattice boltzmann model for simulating immiscible two-phase flows. *Journal of Physics A: Mathematical and Theoretical*, 40(14):4033, 2007.

[15] Xiaowen Shan and Hudong Chen. Lattice boltzmann model for simulating flows with multiple phases and components. *Phys. Rev. E*, 47:1815–1819, Mar 1993.

[16] U. Frisch, B. Hasslacher, and Y. Pomeau. Lattice-gas automata for the navier-stokes equation. *Phys. Rev. Lett.*, 56:1505–1508, Apr 1986.

[17] G. R. McNamara and G. Zanetti. Use of the Boltzmann Equation to Simulate Lattice-Gas Automata. *Phys. Rev. Lett.*, 61:2332–2335, Nov 1988.

[18] F. J. Higuera and J. Jimnez. Boltzmann Approach to Lattice Gas Simulations. *EPL (Europhysics Letters)*, 9(7):663, 1989.

[19] S. Ubertini, P. Asinari, and S. Succi. Three ways to lattice Boltzmann: A unified time-marching picture. *Phys. Rev. E*, 81:016311, Jan 2010.

[20] M. O. Deville, P. F. Fischer, and E. H. Mund. *High-Order Methods for Incompressible Fluid Flow.* Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2002.

[21] M. Griebel, T. Dornseifer, and T. Neunhoeffer. *Numerical Simulation in Fluid Dynamics: A Practical Introduction.* Mathematical Modeling and Computation. SIAM, 1998.

[22] K. Versteeg, H. and W. Malalasekera. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method.* Addison-Wesley, Reading, 1995.

[23] The Finite Element Method for Fluid Dynamics. In O.C. Zienkiewicz, R.L. Taylor, , and P. Nithiarasu, editors, *The Finite Element Method for Fluid Dynamics (Seventh Edition)*, pages i –. Butterworth-Heinemann, Oxford, seventh edition edition, 2014.

[24] S. Chapman and T. Cowling. *Mathematical Theory of Nonuniform Gases.* Cambridge University Press, 1964.

[25] Y. Sone. *Kinetic Theory and Fluid Dynamics.* Modeling and Simulation in Science, Engineering and Technology. Springer, 2002.

[26] M. Junk, A. Klar, and L.-S. Luo. Asymptotic analysis of the lattice Boltzmann equation. *J. Comput. Phys.*, 210(2):676–704, December 2005.

[27] P. Asinari. Generalized local equilibrium in the cascaded lattice Boltzmann method. *Phys. Rev. E*, 78(1), July 2008.

[28] H. Grad. On the kinetic theory of rarefied gases. *Comm. Pure Appl. Math*, 2(4):331–407, 1949.

[29] M. Geier, A. Greiner, and J. G. Korvink. A factorized central moment lattice Boltzmann method. *The European Physical Journal Special Topics*, 171(1):55–61, April 2009.

[30] H. Chen, S. Chen, and W. H. Matthaeus. Recovery of the Navier-Stokes equations using a lattice-gas Boltzmann method. *Phys. Rev. A*, 45(8):R5339, 1992.

[31] Y. H. Qian, D. d'Humières, and P. Lallemand. Lattice BGK Models for Navier-Stokes Equation. *Europhys. Lett.*, 17(6):479, 1992.

[32] P. L. Bhatnagar, E. P. Gross, and M. Krook. A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral One-Component Systems. *Physical Review*, 94(3):511, 1954.

[33] X. He and L.-S. Luo. Theory of the lattice boltzmann method: From the boltzmann equation to the lattice boltzmann equation. *Phys. Rev. E*, 56:6811–6817, Dec 1997.

[34] X. He, X. Shan, and G. D. Doolen. Discrete Boltzmann equation model for nonideal gases. *Physical Review E*, 57(1):R13, 1998.

[35] X. Shan, X.-F. Yuan, and H. Chen. Kinetic theory representation of hydrodynamics: a way beyond the navierstokes equation. *Journal of Fluid Mechanics*, 550:413441, 2006.

[36] D. d'Humières. Generalized Lattice-Boltzmann Equations. In *Rarefied Gas Dynamics: Theory and Simulations, Progress in Astronautics and Aeronautics*, volume 159, pages 450–458. American Institute of Aeronautics and Astronautics, 1992.

[37] P. Lallemand and L.-S. Luo. Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance, and stability. *Phys. Rev. E*, 61(6):6546, 2000.

[38] I. Ginzburg and D. d'Humières. Multireflection boundary conditions for lattice Boltzmann models. *Phys. Rev. E*, 68(6), December 2003.

[39] P. J. Dellar. Incompressible limits of lattice Boltzmann equations using multiple relaxation times. *J. Comput. Phys.*, 190(2):351–370, 2003.

[40] L.-S. Luo, W. Liao, X. Chen, Y. Peng, and W. Zhang. Numerics of the lattice boltzmann method: Effects of collision models on the lattice boltzmann simulations. *Phys. Rev. E*, 83:056710, May 2011.

[41] I. Ginzburg. Equilibrium-type and link-type lattice Boltzmann models for generic advection and anisotropic-dispersion equation. *Advances in Water Resources*, 28(11):1171 – 1195, 2005.

[42] S. Ansumali and I. V. Karlin. Single relaxation time model for entropic lattice Boltzmann methods. *Phys. Rev. E*, 65(5), May 2002.

[43] S. Ansumali and I. V. Karlin. Stabilization of the lattice boltzmann method by the H theorem: A numerical test. *Phys. Rev. E*, 62:7999–8003, Dec 2000.

[44] P. Asinari and I. V. Karlin. Generalized Maxwell state and h theorem for computing fluid flows using the lattice Boltzmann method. *Phys. Rev. E*, 79:036703, Mar 2009.

[45] N. I. Prasianakis, I. V. Karlin, J. Mantzaras, and K. B. Boulouchos. Lattice boltzmann method with restored galilean invariance. *Phys. Rev. E*, 79:066702, Jun 2009.

[46] M. Geier. *Ab initio derivation of the cascaded Lattice Boltzmann Automaton.* PhD dissertation, University of Freiburg - IMTEK, 2006.

[47] M. Geier, A. Greiner, and J. G. Korvink. Cascaded digital lattice Boltzmann automata for high Reynolds number flow. *Phys. Rev. E*, 73(6), June 2006.

[48] M. Geier, M. Schönherr, A. Pasquali, and M. Krafczyk. The cumulant lattice Boltzmann equation in three dimensions: Theory and validation. *Comput. Math. Appl.*, 70(4):507–547, August 2015.

[49] M. Rohde, D. Kandhai, J. J. Derksen, and H. E. A. van den Akker. A generic, mass conservative local grid refinement technique for lattice-Boltzmann schemes. *Int. J. Numer. Meth. Fluids*, 51(4):439–468, June 2006.

[50] H. Chen, O. Filippova, J. Hoch, K. Molvig, R. Shock, C. Teixeira, and R. Zhang. Grid refinement in lattice Boltzmann methods based on volumetric formulation. *Physica A*, 362(1):158–167, March 2006.

[51] M. Schönherr, K. Kucher, M. Geier, M. Stiebler, S. Freudiger, and M. Krafczyk. Multi-thread implementations of the lattice boltzmann method on non-uniform grids for CPUs and GPUs. *Comput. Math. Appl.*, 61(12):3730 – 3743, 2011.

[52] O. Filippova and D. Hänel. Grid Refinement for Lattice-BGK Models. *J. Comput. Phys.*, 147(1):219 – 228, 1998.

[53] A. Dupuis and B. Chopard. Theory and applications of an alternative lattice Boltzmann grid refinement algorithm. *Phys. Rev. E*, 67(6), June 2003.

[54] D. Lagrava, O. Malaspinas, J. Latt, and B. Chopard. Advances in multi-domain lattice Boltzmann grid refinement. *J. Comput. Phys.*, 231(14):4808–4822, May 2012.

[55] E. J. Falagkaris, D. M. Ingram, I. M. Viola, and K. Markakis. PROTEUS: A coupled iterative force-correction immersed-boundary multi-domain cascaded lattice boltzmann solver. *Comput. Math. Appl.*, 74(10).

[56] E. J. Falagkaris, D. M. Ingram, K. Markakis, and I. M. Viola. PROTEUS: A coupled iterative force-correction immersed-boundary cascaded

lattice boltzmann solver for moving and deformable boundary applications. *Comput. Math. Appl.*, 75(4):1330 – 1354, 2018.

[57] Q. Zou and X. He. On pressure and velocity boundary conditions for the lattice Boltzmann BGK model. *Phys. Fluids*, 9(6):1591–1598, 1997.

[58] T. Inamuro, M. Yoshino, and F. Ogino. A nonslip boundary condition for lattice boltzmann simulations. *Physics of Fluids*, 7(12):2928–2930, 1995.

[59] J. Latt and B. Chopard. Lattice Boltzmann method with regularized pre-collision distribution functions. *Math. Comput. Simulation*, 72(2):165–168, 2006.

[60] P. A. Skordos. Initial and boundary conditions for the lattice boltzmann method. *Phys. Rev. E*, 48:4823–4842, Dec 1993.

[61] S. Ansumali and I. V. Karlin. Kinetic boundary conditions in the lattice boltzmann method. *Phys. Rev. E*, 66:026311, Aug 2002.

[62] I. Halliday, L. A. Hammond, and C. M. Care. Enhanced closure scheme for lattice Boltzmann equation hydrodynamics. *Journal of Physics A: Mathematical and General*, 35(12):L157, 2002.

[63] M. Bouzidi, M. Firdaouss, and P. Lallemand. Momentum transfer of a boltzmann-lattice fluid with boundaries. *Physics of Fluids*, 13(11):3452–3459, 2001.

[64] C. S. Peskin. Numerical Analysis of Blood Flow in the Heart. *J. Comput. Phys.*, 25(3):220–252, 1977.

[65] C. S. Peskin. The immersed boundary method. *Acta Numerica*, 11, January 2002.

[66] R. Glowinski, T.W. Pan, T.I. Hesla, D.D. Joseph, and J. Périaux. A Fictitious Domain Approach to the Direct Numerical Simulation of Incompressible Viscous Flow past Moving Rigid Bodies: Application to Particulate Flow. *J. Comput. Phys.*, 169(2):363–426, May 2001.

[67] K. Höfler and S. Schwarzer. Navier-Stokes simulation with constraint forces: Finite-difference method for particle-laden flows and complex geometries. *Phys. Rev. E*, 61(6):7146, 2000.

[68] N. A. Patankar, P. Singh, D. D. Joseph, R. Glowinski, and T.-W. Pan. A new formulation of the distributed Lagrange multiplier/fictitious domain method for particulate flows. *Int. J. Multiphase Flow*, 26(9):1509–1524, 2000.

[69] M.-C. Lai and C. S. Peskin. An Immersed Boundary Method with Formal Second-Order Accuracy and Reduced Numerical Viscosity. *J. Comput. Phys.*, 160(2):705–719, May 2000.

[70] E.M. "Saiki and S." Biringen. Numerical simulation of a cylinder in uniform flow: Application of a virtual boundary method. *J. Comput. Phys.*, 123(2):450 – 465, 1996.

[71] D. "Goldstein, R. Handler, and L." Sirovich. Modeling a No-Slip Flow Boundary with an External Force Field. *J. Comput. Phys.*, 105(2):354 – 366, 1993.

[72] J. Mohd-Yusof. Combined immersed boundaries/b-spline methods for simulations of flows in complex geometries. In *CTR Annual Research Briefs*, page 317327. NASA Ames/Stanford University.

[73] E.A. " Fadlun, R. Verzicco, P. Orlandi, and J." Mohd-Yusof. Combined Immersed-Boundary Finite-Difference Methods for Three-Dimensional Complex Flow Simulations. *J. Comput. Phys.*, 161(1):35 – 60, 2000.

[74] J. "Kim, D. Kim, and H." Choi. An Immersed-Boundary Finite-Volume Method for Simulations of Flow in Complex Geometries. *J. Comput. Phys.*, 171(1):132 – 150, 2001.

[75] E. Balaras. Modeling complex boundaries using an external force field on fixed cartesian grids in large-eddy simulations. *Comput. & Fluids*, 33(3):375 – 404, 2004.

[76] J.-I. Choi, R. C. Oberoi, J. R. Edwards, and J. A. Rosati. An immersed boundary method for complex incompressible flows. *J. Comput. Phys.*, 224(2):757 – 784, 2007.

[77] G. Iaccarino and R. Verzicco. Immersed boundary technique for turbulent flow simulations. *Appl. Mech. Rev.*, 56(3):331–347, 05 2003.

[78] R. Ghias, R. Mittal, and H. Dong. A sharp interface immersed boundary method for compressible viscous flows. *J. Comput. Phys.*, 225(1):528 – 553, 2007.

[79] S. K. Kang and Y. A. Hassan. A comparative study of direct-forcing immersed boundary-lattice Boltzmann methods for stationary complex boundaries. *Int. J. Numer. Meth. Fluids*, 66(9):1132–1158, July 2011.

[80] A.L.F. Lima E Silva, A. Silveira-Neto, and J.J.R. Damasceno. Numerical simulation of two-dimensional flows over a circular cylinder using the immersed boundary method. *J. Comput. Phys.*, 189(2):351 – 370, 2003.

[81] M. Uhlmann. An immersed boundary method with direct forcing for the simulation of particulate flows. *J. Comput. Phys.*, 209(2):448 – 476, 2005.

[82] S.-W. "Su, M.-C. Lai, and C.-A." Lin. An immersed boundary technique for simulating complex flows with rigid boundary. *Comput. & Fluids*, 36(2):313 – 324, 2007.

[83] D.V. Le, B.C. Khoo, and K.M. Lim. An implicit-forcing immersed boundary method for simulating viscous flows in irregular domains. *Computer Methods in Applied Mechanics and Engineering*, 197(25):2119 – 2130, 2008. Immersed Boundary Method and Its Extensions.

[84] K. "Luo, J. Wang, and K." Cen. Full-scale solutions to particle-laden flows: multidirect forcing and immersed boundarymethod. *Phys. Rev. E*, 76, 2007.

[85] Z. Wang, J. Fan, and K. Luo. Combined multi-direct forcing and immersed boundary method for simulating flows with moving particles. *International Journal of Multiphase Flow*, 34(3):283 – 302, 2008.

[86] K. Suzuki and T. Inamuro. Effect of internal mass in the simulation of a moving body by the immersed boundary method. *Comput. & Fluids*, 49(1):173 – 187, 2011.

[87] J. J. Derksen A. ten Cate, C. H. Nieuwstad and H. E. A. Van den Akker. Particle imaging velocimetry experiments and lattice-Boltzmann simulations on a single sphere settling under gravity. *Physics of Fluids*, 14(4), 2002.

[88] Z.-G. Feng and E. E. Michaelides. Robust treatment of no-slip boundary condition and velocity updating for the lattice-Boltzmann simulation of particulate flows. *Comput. & Fluids*, 38(2):370 – 381, 2009.

[89] A. J. C. Ladd and R. Verberg. Lattice-boltzmann simulations of particle-fluid suspensions. *Journal of Statistical Physics*, 104(5):1191–1251, Sep 2001.

[90] L. Shen, E.-S. Chan, and P. Lin. Calculation of hydrodynamic forces acting on a submerged moving object using immersed boundary method. *Comput. & Fluids*, 38(3):691 – 702, 2009.

[91] L.-S. Luo. *Lattice-Gas Automata and Lattice Boltzmann Equations for Two-Dimensional Hydrodynamics.* PhD thesis, Georgia Institute of Technology, 1993.

[92] U. Frisch, D. d'Humieres, B. Hasslacher, P. Lallemand, Y. Pomeau, and J.P. Rivet. Lattice gas hydrodynamics in two and three dimensions. *Compl. Syst. 1*.

[93] X. Shan and H. Chen. Simulation of nonideal gases and liquid-gas phase transitions by the lattice boltzmann equation. *Phys. Rev. E*, 49:2941–2948, Apr 1994.

[94] K. N. Premnath and S. Banerjee. Incorporating forcing terms in cascaded lattice Boltzmann approach by method of central moments. *Phys. Rev. E*, 80(3):036702, 2009.

[95] N. S. Martys, X. Shan, and H. Chen. Evaluation of the external force term in the discrete boltzmann equation. *Phys. Rev. E*, 58:6855–6857, Nov 1998.

[96] Z. Guo, C. Zheng, and B. Shi. Discrete lattice effects on the forcing term in the lattice Boltzmann method. *Phys. Rev. E*, 65(4), April 2002.

[97] A.L. Kupershtokh, D.A. Medvedev, and D.I. Karpov. On equations of state in a lattice Boltzmann method. *Comput. Math. Appl.*, 58(5):965 – 974, 2009. Mesoscopic Methods in Engineering and Science.

[98] Y. Cheng and J. Li. Introducing unsteady non-uniform source terms into the lattice Boltzmann model. *Int. J. Num. Meth. Fluids*, 56(6):629–641, February 2008.

[99] Z.-G. Feng and E. E. Michaelides. The immersed boundary-lattice Boltzmann method for solving fluid - particles interaction problems. *J. Comput. Phys.*, 195(2):602–628, April 2004.

[100] A. Dupuis, P. Chatelain, and P. Koumoutsakos. An immersed boundary-lattice-Boltzmann method for the simulation of the flow past an impulsively started cylinder. *J. Comput. Phys.*, 227(9):4486–4498, April 2008.

[101] X.D. Niu, C. Shu, Y.T. Chew, and Y. Peng. A momentum exchange-based immersed boundary-lattice Boltzmann method for simulating incompressible viscous flows. *Phys. Lett. A*, 354(3):173–182, May 2006.

[102] T. Inamuro. Lattice Boltzmann methods for moving boundary flows. *Fluid Dyn. Res.*, 44(2):024001, April 2012.

[103] C. Zhang, Y. Cheng, L. Zhu, and J. Wu. Accuracy improvement of the immersed boundary-lattice Boltzmann coupling scheme by iterative force correction. *Comput. & Fluids*, 124:246–260, January 2016.

[104] J. Wu and C. Shu. Implicit velocity correction-based immersed boundary-lattice Boltzmann method and its applications. *J. Comput. Phys.*, 228(6):1963–1979, April 2009.

[105] K. N. Premnath, M. J. Pattison, and S. Banerjee. Generalized lattice boltzmann equation with forcing term for computation of wall-bounded turbulent flows. *Phys. Rev. E*, 79:026703, Feb 2009.

[106] D. Ricot, S. Mari, P. Sagaut, and C. Bailly. Lattice boltzmann method with selective viscosity filter. *J. Comput. Phys.*, 228(12):4478 – 4490, 2009.

[107] N. Pellerin, S. Leclaire, and M. Reggio. An implementation of the Spalart - Allmaras turbulence model in a multi-domain lattice Boltzmann method for solving turbulent airfoil flows. *Comput. Math. Appl.*, 70(12):3001–3018, December 2015.

[108] J. Tölke and M. Krafczyk. Second order interpolation of the flow field in the lattice boltzmann method. *Comput. Math. Appl.*, 58(5):898 – 902, 2009. Mesoscopic Methods in Engineering and Science.

[109] M. Geier, A. Greiner, and J. G. Korvink. Bubble functions for the lattice Boltzmann method and their application to grid refinement. *The European Physical Journal Special Topics*, 171(1):173–179, April 2009.

[110] P. Qumr, P. Sagaut, and V. Couailler. A new multidomain/multiresolution method for largeeddy simulation. *Int. J. Numer. Meth. Fluids*, 36(4):391–416.

[111] O. P. Malaspinas. *Lattice Boltzmann Method for the Simulation of Viscoelastic Fluid Flows.* PhD thesis, École polytechnique fédérale de Lausanne, 2009.

[112] M. B. Schlaffer and others. *Non-reflecting boundary conditions for the lattice Boltzmann method.* PhD thesis, Universittsbibliothek der TU Mnchen, 2013.

[113] S. Izquierdo and N. Fueyo. Characteristic nonreflecting boundary conditions for open boundaries in lattice boltzmann methods. *Phys. Rev. E*, 78:046707, Oct 2008.

[114] M.M. Tekitek, M. Bouzidi, F. Dubois, and P. Lallemand. Towards perfectly matching layers for lattice boltzmann equation. *Comput. Math. Appl.*, 58(5):903 – 913, 2009. Mesoscopic Methods in Engineering and Science.

[115] E.W.S. Kam, R.M.C. So, and R.C.K. Leung. Lattice Boltzman Method Simulation of Aeroacoustics and Nonreflecting Boundary Conditions. *AIAA Journal*, 45(7):1703–1712, 2018/04/09 2007.

[116] Gauthier Wissocq, Nicolas Gourdain, Orestis Malaspinas, and Alexandre Eyssartier. Regularized characteristic boundary conditions for the lattice-boltzmann methods at high reynolds number flows. *J. Comput. Phys.*, 331:1 – 18, 2017.

[117] H. Xu and P. Sagaut. Analysis of the absorbing layers for the weakly-compressible lattice Boltzmann methods. *J. Comput. Phys.*, 245:14–42, July 2013.

[118] D. J.” ”Bodony. Analysis of sponge zones for computational fluid mechanics. *J. Comput. Phys.*, 212(2):681 – 702, 2006.

[119] T. Krüger, H. Kusumaatmaja, A. Kuzmin, O. Shardt, G. Silva, and E.M. Viggen. *The Lattice Boltzmann Method.* Graduate Texts in Physics. Springer, 2017.

[120] A. Pinelli, I.Z. Naqavi, U. Piomelli, and J. Favier. Immersed-boundary methods for general finite-difference and finite-volume NavierStokes solvers. *J. Comput. Phys.*, 229(24):9073 – 9091, 2010.

[121] X. Yang, X. Zhang, Z. Li, and G.-W. He. A smoothing technique for discrete delta functions with application to immersed boundary method in moving boundary simulations. *J. Comput. Phys.*, 228(20):7821–7836, November 2009.

[122] Y. Cheng, L. Zhu, and C. Zhang. Numerical Study of Stability and Accuracy of the Immersed Boundary Method Coupled to the Lattice Boltzmann BGK Model. *Commun. Comput. Phys.*, 16(01):136–168, July 2014.

[123] G. Le and J. Zhang. Boundary slip from the immersed boundary lattice Boltzmann models. *Phys. Rev. E*, 79:026701, Feb 2009.

[124] T. Krüger, F. Varnik, and D. Raabe. Efficient and accurate simulations of deformable particles immersed in a fluid using a combined immersed boundary lattice boltzmann finite element method. *Comput. Math. Appl.*, 61(12):3485 – 3505, 2011.

[125] Y. Sui, Y.T. Chew, P. Roy, and H.T. Low. A hybrid method to study

flow-induced deformation of three-dimensional capsules. *J. Comput. Phys.*, 227(12):6351 – 6371, 2008.

[126] T. Seta, R. Rojas, K. Hayashi, and A. Tomiyama. Implicit-correction-based immersed boundary˘lattice Boltzmann method with two relaxation times. *Phys. Rev. E*, 89:023307, Feb 2014.

[127] Y. Cheng and H. Zhang. Immersed boundary method and lattice Boltzmann method coupled FSI simulation of mitral leaflet flow. *Comput. & Fluids*, 39(5):871–881, May 2010.

[128] D. Lycett-Brown and K. H. Luo. Multiphase cascaded lattice Boltzmann method. *Comput. Math. Appl.*, 67(2):350 – 362, 2014. Mesoscopic Methods for Engineering and Science (Proceedings of ICMMES-2012, Taipei, Taiwan, 2327 July 2012).

[129] L. Fei and K. H. Luo. Consistent forcing scheme in the cascaded lattice boltzmann method. *Phys. Rev. E*, 96:053307, Nov 2017.

[130] K. Suzuki and T. Inamuro. A higher-order immersed boundary-lattice boltzmann method using a smooth velocity field near boundaries. *Comput. & Fluids*, 76:105 – 115, 2013.

[131] Tiwari A. and Vanka S. P. A ghost fluid lattice Boltzmann method for complex geometries. *International Journal for Numerical Methods in Fluids*, 69(2):481–498.

[132] H. Mansy, P.-M. Yang, and D. R. Williams. Quantitative measurements of three-dimensional structures in the wake of a circular cylinder. *J. Fluid Mech.*, 270(-1):277, July 1994.

[133] D. Barkley and R. D. Henderson. Three-dimensional Floquet stability analysis of the wake of a circular cylinder. *J. Fluid Mech.*, 322(-1):215, September 1996.

[134] S. C. R. Dennis and G.-Z. Chang. Numerical solutions for steady flow past a circular cylinder at Reynolds numbers up to 100. *J. Fluid Mech.*, 42(03):471, July 1970.

[135] B. Fornberg. A numerical study of steady viscous flow past a circular cylinder. *J. Fluid Mech.*, 98(04):819–855, 1980.

[136] X. He and G. Doolen. Lattice Boltzmann Method on Curvilinear Coordinates System: Flow around a Circular Cylinder. *J. Comput. Phys.*, 134(2):306 – 315, 1997.

[137] F. Nieuwstadt and H. B. Keller. Viscous flow past circular cylinders. *Comput. & Fluids*, 1(1):59–71, 1973.

[138] C. H. K. Williamson. Oblique and parallel modes of vortex shedding in the wake of a circular cylinder at low Reynolds numbers. *J.Fluid Mech.*, 206:579–627, 1989.

[139] C. Liu, X. Zheng, and C.H. Sung. Preconditioned Multigrid Methods for Unsteady Incompressible Flows. *J. Comput. Phys.*, 139(1):35 – 57, 1998.

[140] S. Mittal and T. E. Tezduyar. Massively parallel finite element computation of incompressible flows involving fluid-body interactions. *Comput. Methods Appl. Mech. Engrg.*, 112(1-4):253–282, 1994.

[141] A.A. Johnson and T.E. Tezduyar. Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces. *Comput. Methods Appl. Mech. Engrg.*, 119(1):73 – 94, 1994.

[142] C.-C. Liao, Y.-W. Chang, C.-A. Lin, and J.M. McDonough. Simulating flows with moving rigid boundary using immersed-boundary method. *Comput. & Fluids*, 39(1):152 – 167, 2010.

[143] D. Kim and H. Choi. Immersed boundary method for flow around an arbitrarily moving body. *J. Comput. Phys.*, 212(2):662 – 680, 2006.

[144] J.R. Morison, J.W. Johnson, and S.A. Schaaf. The Force Exerted by Surface Waves on Piles. *J. Petrol. Technol.*, 2:149–154, 1950.

[145] B. Uzunoğlu, M. Tan, and W. G. Price. Low-Reynolds-number flow around an oscillating circular cylinder using a cell viscousboundary element method. *Int. J. Numer. Meth. in Engng*, 50(10):2317–2338, 2001.

[146] R. Yuan, C. Zhong, and He Zhang. An immersed-boundary method based on the gas kinetic bgk scheme for incompressible viscous flow. *J. Comput. Phys.*, 296:184 – 208, 2015.

[147] R. Godoy-Diana, C. Marais, J.-L. Aider, and J.-E. Wesfreid. A model for the symmetry breaking of the reverse Bénardvon Kármán vortex street produced by a flapping foil. *J. Fluid Mech.*, 622:2332, 2009.

[148] R. E. Radespiel, J. Windte, and U. Scholz. Numerical and Experimental Flow Analysis of Moving Airfoils with Laminar Separation Bubbles. *AIAA Journal*, 45(6):1346–1356, 2007.

[149] W. Yuan, M. Khalid, J. Windte, U. Scholz, and R. Radespiel. *An Investigation of Low-Reynolds-Number Flows Past Airfoils.* AIAA, 2017/06/25 2005.

[150] M. Galbraith and M. Visbal. *Implicit Large Eddy Simulation of Low-Reynolds-Number Transitional Flow Past the SD7003 Airfoil.* AIAA, 2017/06/25 2010.

[151] Ismail H. Tuncer and Mustafa Kaya. Thrust generation caused by flapping airfoils in a biplane configuration. *Journal of Aircraft*, 40(3):509–515, 2017/07/03 2003.

[152] M.M. Murray and L.E. Howle. Spring stiffness influence on an oscillating propulsor. *Journal of Fluids and Structures*, 17(7):915 – 926, 2003.

[153] J.-M. Miao and M.-H. Ho. Effect of flexure on aerodynamic propulsive efficiency of flapping flexible airfoil. *Journal of Fluids and Structures*, 22(3):401 – 419, 2006.

[154] D.A. Read, F.S. Hover, and M.S. Triantafyllou. Forces on oscillating foils for propulsion and maneuvering. *Journal of Fluids and Structures*, 17(1):163 – 183, 2003.

[155] Dennis S.C.R. and Chang G.-Z. Numerical solutions for steady flow past a circular cylinder at reynolds numbers up to 100. *J. Fluid Mech.*, 42(3):471–489.

[156] S. Sen, S. Mittal, and G. Biswas. Steady separated flow past a circular cylinder at low reynolds numbers. *J. Fluid Mech.*, 620:89119, 2009.

[157] Park J.Y., Choi H. C., and Kwon K. Y. Numerical solutions of flow past a circular cylinder at reynolds numbers up to 160. *KSME International Journal*, 12(6):1200–1205.

[158] Sharman B., Lien F. S., Davidson L., and Norberg C. Numerical predictions of low reynolds number flows over two tandem circular cylinders. *International Journal for Numerical Methods in Fluids*, 47(5):423–447.

[159] G. H. Koopmann. The vortex wakes of vibrating cylinders at low reynolds numbers. *J. Fluid Mech.*, 28(3):501512, 1967.

[160] A. Placzek, J.-F. Sigrist, and A. Hamdouni. Numerical simulation of an oscillating cylinder in a cross-flow at low reynolds number: Forced and free oscillations. *Comput. & Fluids*, 38(1):80 – 100, 2009.

[161] P. Anagnostopoulos. Numerical study of the flow past a cylinder excited transversely to the incident stream. Part 1: Lock-in zone, hydrodynamic forces and wake geometry. *J. Fluids Struct.*, 14(6):819 – 851, 2000.

[162] J. D. Eldredge. Numerical simulation of the fluid dynamics of 2d rigid body motion with the vortex particle method. *J. Comput. Phys.*, 221(2):626 – 648, 2007.

[163] Z. J. Wang, J. M. Birch, and M. H. Dickinson. Unsteady forces and flows in low Reynolds number hovering flight: two-dimensional computations vs robotic wing experiments. *Journal of Experimental Biology*, 207(3):449–460, 2004.

[164] S. Chen, C. Peng, Y. Teng, L.-P. Wang, and K. Zhang. Improving lattice Boltzmann simulation of moving particles in a viscous flow using local grid refinement. *Comput. & Fluids*, 136:228 – 246, 2016.

[165] Y. Kuwata and K. Suga. Imbalance-correction grid-refinement method for lattice Boltzmann flow simulations. *J. Comput. Phys.*, 311:348 – 362, 2016.

[166] M. Stiebler, M. Krafczyk, S. Freudiger, and M. Geier. Lattice boltzmann large eddy simulation of subcritical flows around a sphere on non-uniform grids. *Comput. Math. Appl.*, 61(12):3475 – 3484, 2011. Mesoscopic Methods for Engineering and Science Proceedings of ICMMES-09Mesoscopic Methods for Engineering and Science.

[167] K. N. Premnath, M. J. Pattison, and S. Banerjee. Dynamic subgrid scale modeling of turbulent flows using lattice-boltzmann method. *Physica A: Statistical Mechanics and its Applications*, 388(13):2640 – 2658, 2009.

[168] A. Pradhan and S. Yadav. Large Eddy Simulation Using Lattice Boltzmann Method Based on Sigma Model. *Procedia Engineering*, 127:177 – 184, 2015. INTERNATIONAL CONFERENCE ON COMPUTATIONAL HEAT AND MASS TRANSFER (ICCHMT) - 2015.

[169] M. Weickert, G. Teike, O. Schmidt, and M. Sommerfeld. Investigation of the LES WALE turbulence model within the lattice Boltzmann framework.

Comput. Math. Appl., 59(7):2200 – 2214, 2010. Mesoscopic Methods in Engineering and Science.

[170] O. Malaspinas and P. Sagaut. Advanced large-eddy simulation for lattice Boltzmann methods: The approximate deconvolution model. *Physics of Fluids*, 23(10):105103, 2011.

[171] A. Schneider. *A Consistent Large Eddy Approach for LatticeBoltzmann Methods and its Application to Complex Flows.* PhD dissertation, 2015.

[172] J. Latt. *Hydrodynamic limit of lattice Boltzmann equations.* PhD dissertation, Université de Genéve, 2007.

[173] D. Conrad. *A Viscosity Adaptive Lattice Boltzmann Method.* PhD dissertation, TU Kaiserslautern, 2015.

[174] D. J. Holdych, D. R. Noble, J. G. Georgiadis, and R. O. Buckius. Truncation error analysis of lattice Boltzmann methods. *J. Comput. Phys.*, 193(2):595 – 619, 2004.

[175] "O. Filippova and D. Hänel ". Boundary-Fitting and Local Grid Refinement for Lattice BGK Models. *International Journal of Modern Physics C*, 09(08):1271 – 1279, 1998.

[176] S. M. Guzik, T.H. Weisgraber, P. Colella, and B. J. Alder. Interpolation methods and the accuracy of lattice-Boltzmann mesh refinement. *J. Comput. Phys.*, 259:461 – 487, 2014.

[177] I. V. Karlin, A. Ferrante, and H. C. Öttinger. Perfect entropy functions of the Lattice Boltzmann method. *Europhys. Lett.*, 47(2), 1999.

[178] N. Frapolli, S. S. Chikatamarla, and I. V. Karlin. Entropic lattice Boltzmann model for compressible flows. *Phys. Rev. E*, 92:061301, Dec 2015.

[179] Ali Mazloomi M., Shyam S. Chikatamarla, and Iliya V. Karlin. Entropic lattice Boltzmann method for multiphase flows: Fluid-solid interfaces. *Phys. Rev. E*, 92:023308, Aug 2015.

[180] B. Dorschner, S.S. Chikatamarla, F. Bsch, and I.V. Karlin. Grad's approximation for moving and stationary walls in entropic lattice Boltzmann simulations. *J. Comput. Phys.*, 295:340 – 354, 2015.

[181] S.S. Chikatamarla and I.V. Karlin. Entropic lattice Boltzmann method for turbulent flow simulations: Boundary conditions. *Physica A: Statistical Mechanics and its Applications*, 392(9):1925 – 1930, 2013.

[182] M. Atif, P.K. Kolluru, C. Thantanapally, and S. Ansumali. Essentially entropic lattice boltzmann model. *Phys. Rev. Lett.*, 119:240602, Dec 2017.

[183] C. Chang, C.-H. Liu, and C.-A. Lin. Boundary conditions for lattice boltzmann simulations with complex geometry flows. *Comput. Math. Appl.*, 58(5):940 – 949, 2009. Mesoscopic Methods in Engineering and Science.

[184] W. Zhao and W.-A. Yong. Single-node second-order boundary schemes for the lattice Boltzmann method. *J. Comput. Phys.*, 329:1 – 15, 2017.

[185] W.-A. Yong, W. Zhao, and L.-S. Luo. Theory of the Lattice Boltzmann method: Derivation of macroscopic equations via the Maxwell iteration. *Phys. Rev. E*, 93:033310, Mar 2016.

[186] N. M. Olson. A near-boundary modification for the link bounce-back boundary condition in the lattice boltzmann method. *J. Comput. Phys.*, 301:102 – 110, 2015.

[187] X. Yin and J. Zhang. An improved bounce-back scheme for complex boundary conditions in lattice Boltzmann method. *J. Comput. Phys.*, 231(11):4295 – 4303, 2012.

[188] J. C.G. Verschaeve and B. Mller. A curved no-slip boundary condition for the lattice Boltzmann method. *J. Comput. Phys.*, 229(19):6781 – 6803, 2010.

[189] H.C. Lee, S. Bawazeer, and A.A. Mohamad. Boundary conditions for lattice Boltzmann method with multispeed lattices. *Comput. & Fluids*, 162:152 – 159, 2018.

[190] R. W. Nash, H. B. Carver, M. O. Bernabeu, J. Hetherington, D. Groen, T. Krüger, and P. V. Coveney. Choice of boundary condition for lattice-Boltzmann simulation of moderate-Reynolds-number flow in complex domains. *Phys. Rev. E*, 89:023303, Feb 2014.

[191] Martin Geier, Andrea Pasquali, and Martin Schnherr. Parametrization of the cumulant lattice Boltzmann method for fourth order accurate diffusion part i: Derivation and validation. *J. Comput. Phys.*, 348:862 – 888, 2017.

[192] Z. Li, J. Favier, U.o D'Ortona, and S. Poncet. An immersed boundary-lattice Boltzmann method for single- and multi-component fluid flows. *J. Comput. Phys.*, 304:424 – 440, 2016.